



Development of Web-based Database System for Publications and Thesis

Submitted to the University of Cyprus as partial completion of the requirements for a
B.Sc. Degree in Electrical and Computer Engineering

Department of Electrical and Computer Engineering
May 2017

by Christos Katomoniatis

Development of Web-based Database System for Publications and Thesis

Examination Committee

Dr. Maria Michael
Professor, Department of ECE, Advisor

Dr. Theocharis Theocharides
Professor, Department of ECE

ACKNOWLEDGEMENTS

After so long of hard work and I can not believe it is finally finished!

All those people who contributed in my work during that period I would like to thank them all.

First and foremost I would like to thank and express my sincere gratitude and unrestrained appreciation to my supervisor Dr. Maria K. Michael, for her continuous support, patience, motivation and immense knowledge, and providing me with an excellent atmosphere for doing my project work. You are a great professor, advisor and supporter and I really appreciate every second you dedicated to our meetings in order to complete this project.

Furthermore, I would like to thank some of my friends and colleagues who helped me a lot during my thesis. I would like to thank Savvas for providing me a lot of help regarding Pages and also for formatting this report. I would like to thank my friend Marios who helped me in reviewing English language.

Beyond friends, there is family. Thanks for your love and support for no matter what I decided to do, you were always there for me.

Christos Katomoniatis
BSc Student
University of Cyprus, Cyprus
Email: ckatomo1@ucy.ac.cy

ABSTRACT

Technology evolves in the department of computers which leads to the increase of data we have to manage. Thus creating the need of systems that can manage big data in fast and effective ways. In the long run, saving loads of data by creating new systems becomes a necessity for an organisation or an enterprise.

Needless to say that the creation of a system could benefit in the management of data, which also gives the opportunity to read, write, edit, update, delete, search in a database through user friendly interfaces. With a proper design and implementation of database , the data could be fetched easily and fast enough with multiple queries and simultaneously give the ability to modify the data.

In this thesis it is presented the analysis, design and implementation of a system where publications and thesis are going to be posted through a website. Before the implementation of this system, there was no way to archive, manage and report all those types of publications of the faculty and researchers of the ECE department as well as various Diploma, MSc and PhD Theses.

Although the primary purpose was not just to implement this system, but to build a system with correct and understandable code that changes could not affect the design of the database. The implementation of this system ensures the security, access and management of data for authors who submit their publications and gives them the opportunity to create reports of their own.

The system was implemented in PHP, HTML, CSS for the user interface, Javascript for the back-end functions and SQL (Structure Query Language) that was used for the creation and design of the database.

Table of Contents

Chapter 1 – Introduction	7-9
1.1 General	7
1.2 Purpose	8
1.3 Vision	8
1.4 Glossary of terms	9
Chapter 2 – Object Oriented Software Engineering	10-29
2.1 Object Oriented Analysis	10
2.2 Software Development Process	12
2.3 Requirement Engineering	16
2.4 Requirements Analysis	20
2.5 System Architecture	27
Chapter 3 – Modelling and System Specification	30-37
3.1 System Modelling	30
3.2 UML State Machine Diagrams	32
3.3 Relational Database Management System	35
Chapter 4 – Data Representation	38-40
4.1 Description of Database	38
4.2 Apache Server	38
4.3 MySQL System	38
4.4 List of Database Tables	39
Chapter 5 – Design and User Interface	41-48
5.1 Design Principles	41
5.2 System Design	42
5.3 User Interface Design	46
Chapter 6 – Programming Overview	49-57
6.1 PHP	49
6.2 HTML	53
6.3 PHPMyAdmin	53
6.4 Javascript	54
6.5 Solutions to Selected Problems	55

Chapter 7 - Implementation	57-377
7.1 Website Overview	57
7.2 High Level View	58
7.3 Compatibility and Accessibility	59
7.4 Implementation	59
7.5 Code	72
Conclusion	374
Bibliography	375

LIST OF FIGURES

Figure 1 - V Model	13
Figure 2 - Waterfall Model	14
Figure 3 - Use Case Diagram	25
Figure 4 - UML Diagram	32
Figure 5 - Initial Home Page Design	43
Figure 6 - Final Home Page Design	43
Figure 7 - User Interfaces Diagram	46
Figure 8 - Scalable UI Design	47
Figure 9 - Problem Research Diagram	55
Figure 10 - High Level View	58

Chapter 1

Introduction

“The beginning is the most important part of the work”
-Plato-

1.1 General

There are several challenges in today's technological systems and organising and storing data is one of them. Without a doubt the quantities of data now available compared to the quantities of data of the future are exiguous. But what is the key requirements for big data storage?

Obviously, how well it can handle those very large amounts of data, performance, security with build, robust quality and well-maintained design. In design and documentation if conflict arises, clarity should be preferred to precision because, as will be described, the key problem of software development is having a functioning communication between the involved human parties.

The developer should always keep in mind that the software is written for people and not for computers. Computers just run this software which is a minor point in my perspective. It is people who understand, maintain, improve and use the software to solve the problems that might occur.

Software engineering is solving a problem by an effective abstraction and representation. The existing technologies evolve or become obsolete, but the underlying principles and concepts will likely resurface in new technologies. A single developer might not be able to solve a complex problem.

With software engineering, is not consisted with only a developer but also with quality assurance tester, system architect, platform engineer, costumer, project manager. Writing the code is not just the implementation, it involves the guidelines and writing documentation and also writing unit tests. Those unit tests should fit together in order

to spot the problematic areas using metrics and furthermore improve the quality there.

1.2 Purpose

In this thesis is presented the analysis, design and implementation of an information Web-based management system , which ensures security, access and management of data for authors/users and admins. Before the implementation of this system, publications and theses were a sea of unorganised data which took time and effort to find, handle, store or even search. Therefore this system would be exactly what is needed in order to daily monitor and process loads of data.

More specifically the system can add, edit, update, delete, import, export, search and fetch data through-out 5 tables that are stored in MySQL database. The system has been implemented in PHP,HTML,Javascript Languages, Web services, Apache Server for the website. Additionally the MySQL Language (Structure Query Language) used to create the databases.

1.3 Vision

What is required in this system is speed, simplicity and security. Envision being able to search for a particular author or journal or even a date and edit or update the database within a click.

Computer technology through the years has greatly introduced the idea of automation and storing large data as well as being a learning tool for those who seek it. Nowadays everyone has access to the internet in some way, thus to allow for effortless accessibility to the common use, requiring a web based software service.

1.4 Glossary of Terms

Database: an organised collection of data that are typically organised to model relevant aspects of reality in a way that supports process rewriting this information.

Algorithms: a step-by step procedure for calculations.

Preview List: a list of tracked contracts or projects stored within the system for a particular registered user.

Web service: is a method of communicating between two electronic devices over the World Wide Web.

SQL: is a computer language databases, designed for data management in a management system relational database (RelationalDatabaseManagementSystem, RDBMS), which initially was based on relational algebra. The language includes features retrieval and data update, shapes creation and modification and relational tables, and access control to the data.

SQL Server: It is a relational database management system (RDBMS) from Microsoft that is designed for the business environment.

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

Requirements analysis: (requirements engineering) is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications.

Javascript: is an object-oriented computer programming language commonly used to create interactive effects within web browsers.

Chapter 2

Object Oriented Software Engineering

2.1 Object Oriented Analysis

Object oriented programming methodology based on objects which is concerned to develop an application, a web-based system, on real time, so more emphasis is given on data unlike the other programming styles like structured or functional. The concept of having objects makes OOP more organised, reusable and speedy. Today most of the programming languages support OOP. Most of them have enhanced features to make OOP more easy and maintainable.

So firstly let's analyse an object. An object consists of attributes and methods. Attributes define the properties of the object while methods define its behaviour. In OOP, object is considered as an instance of a class. Considering our web-based system , "Admins" and "Users" can be considered as objects and "Users name" can be considered as their attributes.

Secondly, a class is a simple representation of a type of object. Using the blueprint analogy, a class is a blueprint of an object and an object is a building made from that blueprint. It consists of a name , attributes and methods. Encapsulation , inheritance and polymorphism are the main fundamental concepts in OOP.

Inheritance

Inheritance describes the ability to create new classes based on an existing class. So the object can inherit properties from another object while defining common code only one place with 1 enhancing maintainability of the software. There are many examples that could represent what inheritance means. So for instance we have a "system", which is the parent class. And it is connected with "database" and "website" below the system. Both "database" and "website" inherit all the fields and methods of the parent class "system" . Hence inheritance guarantees code reusability.

Polymorphism

Polymorphism is the ability to take more than one forms depending on data type or class, so that multiple classes can be used interchangeably, even though each class implements the same properties or methods in different ways. Hence polymorphism guarantees maintainability of the software. Interface introduces polymorphism. It contains only definition of methods, properties and events, so the class that implements the interface, has the implementation or declaration to achieve polymorphism.

Advantages of Object Oriented Programming

- Easy to maintain and extend existing code
- Enhanced code reusability
- Object hiding can be achieved
- Improved reliability and flexibility, as objects can be dynamically called and accessed, new objects may be created at any time
- Faster development, as reusing software modules lowers the time usage
- Cost effectiveness, as reusing software modules lowers the cost of development

Disadvantages of Object Oriented Programming

- Need extra time and effort for planning
- Not suitable for all types of problems
- OOP programs are generally larger and slower than normal programs

Web services

Web service is a software system designed to connect to other software applications. An application can publish its function or message to the rest of the world using web services. Web services use XML to code/decode data. Other applications can interact with web service using SOAP messages.

Web services offer many benefits over other types of such systems.

- Interoperability – This is the most important benefit of web services. Web Services allow different applications to interact with each other and share data and services among themselves. For example VB or .NET application can communicate with java web services and vice versa.
- Low cost communication as web services use SOAP over HTTP protocol.
- Reusability
- Usability – Other applications have the freedom to choose the web service they need.

A logical view of web services architecture which is based on three primary roles, service requester, service provider and service registry. Providers of the web services are known as service providers, users of the web services are known as clients or service requester. The essential part of web services is the interact relationship between a service provider and a service requester. These three roles interact using publish, find and bind operations. The service provider publishes the service description in a service registry. The service requester finds the service description in a service registry and uses the information in the description to bind to a service. The service registry provides a centralised location for storing service descriptions. They will become interface to a huge world of data and query services. UDDI, a platform independent, XML based registry is an example of service registry.

2.2 Software Development Process

Software Development Process commonly comprises sequence of work activities, actions, and tasks that are undergone to create the final product. In the context of software development, the final product is a software application (the web-based system), a plug-in component, or a software service solution. However, software development processes are complex and unmanaged process could easily lead to catastrophic failure in delivering a usable system. While there are many factors that contribute to its complexity, the two main reasons described by:

- 1) Intellectual and creative processes rely on people's decisions and judgement and
- 2) The environment may vary hence producing rapidly changing software requirements or strictly defined criteria.

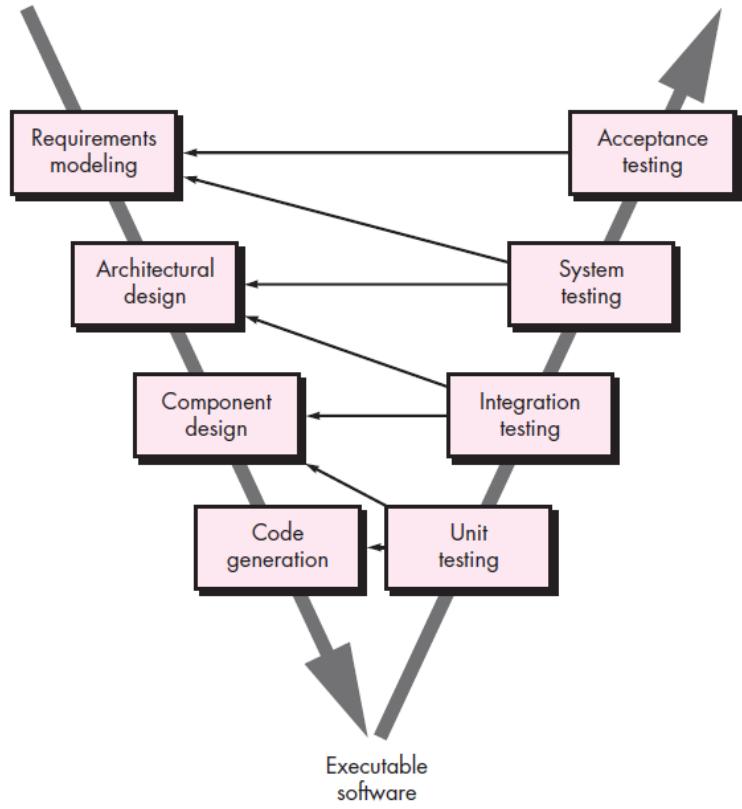
Consequently, careful planning of development activities is required and this results in the adoption of software development process model. A software process model is an abstract representation of interrelated activities in software development . It describes the general approaches in structuring activities and some techniques to produce deliverables. Selecting a suitable model would cut down the development time and increase the quality of the output. Following sections describes several widely applied software process model in the software industry.

Waterfall Model

Waterfall model is a traditional software process model introduced by Royce. It is a rigid and linear document driven methodology. This model is known as the waterfall model because it proceeded from one phase to another in a cascading order. Before each phase can begin, each of the phases has a definite set of deliverables that must be approved by project sponsor or project supervisor, after the stakeholders have elicited them. However, the process of producing and approving these deliverables will incur significant cost. The waterfall model often receives criticism on its inability to accommodate changes because the project freezes system specification upon deliverables sign-off. In a dynamic business environment, it is often difficult for user to state all requirements explicitly. The waterfall model lacks the ability to accommodate natural uncertainty and the changing need of users.

Overview of the waterfall and V models

Figure 1 - V Model



Another serious disadvantage of the waterfall model is that testing is often left to the end of the project. Errors and feedbacks obtained in later stages will require additional effort to resolve. Eventually, this will lead to a software product that not fit for user need. An enhanced variant of the waterfall model known as the V-model has improved to this issue.

Figure at the left illustrates the quality assurance actions associated with deliverables of earlier phases in the V-model. Verification and validation

approaches applied to earlier engineering work could significantly reduce errors found in later stages. However, the V-model does not explicitly describe actions taken in order to deal with errors found during testing. Nevertheless, waterfall model does show its strength when used in project where requirements are well understood and stable during development. Documents produced during each phase provide traceability to address safety and legal issues when such concerns are critical to the user.

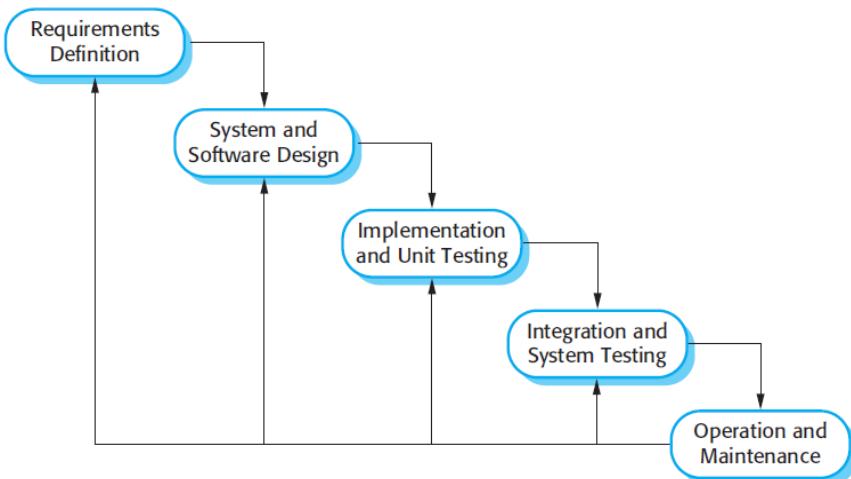


Figure 2 - Waterfall Model

Evolutionary Model

Evolutionary model encapsulates two fundamental approaches: incremental and interactive. When addressing changing requirements. It organises processes in a manner that enables the development of increasingly complete versions of software based on customer feedbacks through a series of iteration. The two fundamental types of evolutionary model that will be covered are prototyping and spiral model.

The idea of prototyping is to enable users to interact and experiment with early prototypes which encapsulate a set of mutual understanding requirements. In [22], prototypes are described as an initial version of a software system used to demonstrate concepts, explore design options, in-depth problems and their possible solutions. Commonly, there are two types of prototypes:

- a. **Common prototypes** aim to explore customer requirements through building an incrementally usable system. Prototypes with a minimum set of basic requirements are built and presented for the customer's evaluation. The prototypes evolves by the implementation of customer proposed features and changes until its functionalities finally agreed by customer
- b. **Throwaway prototypes** aim to gather information and generate ideas on how system should be built. Commonly during project start up, the user may not fully understand their need and the developer may not share understanding on certain features. To clarify these uncertainties, a design prototype which contain just enough details is built for evaluation. Once issues have been clarified, developers could then move on to an actual design and implementation.

Allowing requirements to be implemented rapidly is the key advantage to prototyping. However, this may lead to stakeholder confusion by mistreating what they see as final

version of the system. Stakeholders should be well aware that some prototypes only serve as tools to gather requirements and may vary from the final product.

The spiral model is an evolutionary process model that combines the iterative nature of prototyping while retaining the systematic approaches of waterfall model which was explained earlier. Unlike the waterfall model in which it is hard to backtrack to previous phases once deliverables freeze, spiral model can be adopted throughout system development phases which takes its name from spiral (the widening curve). Explicit recognition of risk in the spiral model is the main difference compared with other process models . However, understanding and mitigating the risks potentially reduces things that can go wrong. Iteration over entire phases of software process would be costly. Hence, the spiral model is more suitable for large-scale projects, which contains high risk and requires well-structured approaches.

Although for projects like web-based database system the spiral model is not very suitable but it definitely requires the same well-strutted approaches.

Agile Development

Agile development processes have emerged to be the dominant software process model in recent years. Agile processes focuses on people, communication, working software, systems and responding to change as opposed to plan-driven models that have high process bureaucracy. Design and implementation are the central activities in agile development process. It would also be possible to incorporate requirements elicitation and testing into these activities, for instance, applying test-driven development (TDD). In TDD, the developer first writes test cases before writing actual implementations. This serves as the preliminary steps to clarify requirements and understanding for problem domains. Developers then code the actual implementations and execute tests to verify the implementations.

Furthermore, a comparison of the three models discussed above based on several concerns that may affect development activities. These concerns, together with their explanations, are listed below:

- a. Requirement elicitation, presents approaches to gather requirements for system
- b. Change management, reflects how changes will be handled throughout project
- c. Validation, explains when testing will be done during project
- d. Delivery discuss how quickly and often the software features will be delivered and
- e. Design modelling covers the depth of design processes during modelling activity.

Based on the comparison, agile development clearly exhibited features that meet the needs. Requirements of user tasks and roles to model intuitive UI. Apparently, user stories of agile development fit better with these requirements. In addition, agile development has factored change management in the model. Its ability to cope with changes reduces the risk of delivering products that does not meet the objectives. The earlier the system is tested, the less effort will be spent on the error that may arise in end of the project. TDD of agile practise embraces this idea and encourages testing done before development. Connected to this, frequent delivery also implies that new enhancement have actually been verified in smaller scale. It reduces complexity by testing only parts that have been changed. Upfront design often leads to design paralysis when the developer tries to adopt concerns and considerations that may not be materialised in the future of project. This is why agile development prefers modelling just enough detail to support an anti-pattern that involves excessive up-front analysis and design but no actual action taken and refactor as required. Finally, lightweight agile process fits well into small-scale development which involves only single developer.

2.3 Requirement Engineering

The techniques utilised to gather and analyse requirements are discussed which then leads to documents on functional and non-functional requirements. These requirements are then translated into informative models to assist decision-making and problem understanding in the Requirement Modelling phase. Finally, the chapter reviews several techniques to manage software development activities that could help to realise these requirements.

The first step of the project is to understand user (and indeed stakeholders) requirements for building this system. Requirements of a system can be defined as descriptions of what services it could provide and the constraints on its operation . These requirements directly address user needs in term of using the system to achieve their business operation – or process.

The process of understanding requirements, known as Requirement Engineering, involves a broad spectrum of processes, tasks and techniques to collect, analyse, document, and verify the requirements. Requirements Engineering is capable of bridging the gap between analysis and design of system as the output of such processes could serve as input to modelling design of systems. It starts with gathering user requirements, then performs analysis on the gathered requirement, and finally documents them in an appropriate format.

Following sections will cover the understanding and application of techniques that would help to formulate user requirement.

Requirement Elicitation

Requirement elicitation are concerns with identifying problems to be solved, what the user (or stakeholders) are trying to accomplish with the system, and how the system addresses the business need or generally the business model. The process begins by understanding and analysing the business problems. Business analysis often reviews that people within an organisation would have different needs (or opinions) and views concerned with the overall requirements of the system. They are stakeholders who either directly interact with or are indirectly affected by the system requirements. Hence, the focuses of requirement elicitation are to analyse the stakeholders' roles and how the operations that they performing – affects the system and hence must be specified in the requirements.

Stakeholder Analysis

Stakeholders are of primary importance to any project due to enormous project resource that has been invested to know exactly what the user wants. If stakeholders are approached earlier in the project, it is easier to communicate their requirement and work out their high priority concerns. The initial step to discover stakeholders requirements would be via Stakeholder Analysis. Stakeholder Analysis views a system as "a complex set of interacting elements which working together to satisfy needs or objectives". The idea is to discover how, when and where stakeholders are involved in the process. The different levels of stakeholders' involvement in the system can be viewed from a stakeholder analysis.

The analysis process starts by identifying the first group of people who form part of the system (in a hierarchical manner) namely the operators. They are referred to Admin and Users. They undertake specific tasks to achieve their operational goals and the system cannot operate without them. Following this, stakeholder analysis continues to discover the next group of stakeholders (in the hierarchy), usually beneficiary of the first group. The admin may be viewed as a direct functional beneficiary who practically can do everything in the system. In additions the user needs to fill in the form and submit it, by logging in with his User ID. Finally the admin has access to edit and manage all the data.

Identifying Stakeholder Operations

After the initial analysis of stakeholders, the next step is to understand the responsibilities of stakeholders. However, precisely identify the operations involved in each distinct roles

is a prerequisite for detailed task analysis in design phases. The functions that support their operations usually are requirements of the system.

It is important to gain insight in to what kind of system should be implemented and the level of change that may affect organisation before determining which requirements are appropriate for a given system. Hence, the steps taken after gathering the initial requirement involve performing an analysis on information obtained. Some of the basic techniques that could be applied during this process as discussed by and are described in the following:

- a. Classification and organisation of requirements, involves grouping related requirements and organises into logical clusters or modules. Using model of system architecture is a common way to discover possible modules (sub-system) and associate related requirements to them.
- b. Prioritisation and negotiations, involves prioritisation requirements resolve conflicting requirements through negotiation with stakeholder. The concern is to achieve a set of agreed requirements that considered views of stakeholder involved.
- c. Additionally, the analysis process may also involves business perspective such as business process automation, business process improvement and business process reengineering. Business process may not too be the focus of this report, but possible improvement and automation should be considered as system requirements.

Requirement Classification and Organisation

As specified above, organising requirements involves grouping requirement into related logical clusters to identify their relationship and dependency. Requirements may be vague at this stage because they are stated in the form of stakeholder's operational goals. Nevertheless, they could be translated into sets of required system functions. As for the complex software system, grouping related system functions often leads to a modularity view. Adopting modularity views in software architecture allows developers to have clear segregation of each subsystem concern and their relationships.

The result of grouping requirements is based on system functions. Each grouping is given a module naming. Additionally, it relates the operators (stakeholders) operation goals specified to system functions. The last concern in the table is the dependencies of each module, serving as important criteria when prioritising the requirements in the next section.

Prioritising Requirement

Prioritising requirement involves ranking requirements by weighting the characteristic of requirements in terms of user needs and dependencies. High priority requirements should be addressed first because other requirements often depend on them. These requirements would also fulfil basic operation goals of stakeholders. Prioritising requirement is an activity according to the principle of incremental planning.

Requirements can be changed depending on the time available and their relative priority.

Based on module dependencies, they should have higher priority compare with other modules. However, each requirement maybe further broken down into sub-requirements for enhanced usability or improved workflow. They may not be essential to basic operations and thus would subject to lower priority. In brief, priority of requirements is concerned with balancing dependencies and user needs.

A method of prioritising requirements in the format of features lists, provides initial abstraction on the important requirements, followed by optional features. The list will require its priority to be further elaborated during requirement specification to achieve optimum ranking. This is described in the functional and non-functional requirement sections.

Requirement Specification

Requirement specification aims to define requirements in clear and unambiguous language based on requirement identified during requirement elicitation and requirement analysis. The requirements are evolved over time and become more accurately reflect the needs of the stakeholders. The next steps are to document these requirements and treat them as ultimate references of domain knowledge and business rules. Documenting requirements can be challenging, and it is aligned to the fact that the software industry appeared to use the term “requirement” inconsistently. Requirement could be a high-level abstract description of system provided services, or comprehensive formal definition of system functional units. Given the vast difference of stakeholder’s perspectives and interests about the best way of specifying requirements, a further investigation into approaches of documenting requirements is necessary.

Requirement documents which are also known as software requirement specifications (SRS), contain important statements describing the software product to be built. The level of details may vary depending on the type of developing system. Safety critical and complex systems often require detailed description of constraints or essential domain knowledge. On the other hand, requirements for commercial software are often changing and become out-of-date quickly. It appears that use case and story card from agile

development methods are more flexible in capturing business requirements. Based on suggestion, the agile approaches in documenting requirement are:

1. Focus on software, not documentation. Create it only if it is essential to the work effort
2. Keep it simple. Create the most minimalist version of each artefact and use simple tools
3. Proceed iteratively. Start by identifying a high-level model and gather the details as the work proceeds
4. Collaboration could improve communication thus reduce need for documentation.

Following on from this, the next steps are looking at two major categories of requirements: functional requirements and non-functional requirements.

Functional Requirement

Functional requirements describe the features that the system has to provide. It often describes the expected features the user can utilise to perform their task. It may cover certain business processes and procedures that the software must follow and perform to achieve user goals, sometimes termed business logic. Thus, Functional Requirements are often associated with desired behaviour characteristics of developing software to produce the expected result. The primary Functional Requirement of the system is to send forms to the database with information that are filled from the user.

2.4 Requirements Analysis

Identifier	Points*	Requirements
REQ1	4	The system shall ensure that all user password shall remain secure
REQ2	3	The system should have loading times no greater than 3 seconds with fair broadband speeds, great performance
REQ3	5	The system should provide the user/admin with options in the menu
REQ4	5	The system can support multiple users and a few admins
REQ5	7	The system should be able to fetch data with a click of a button

Identifier	Points*	Requirements
REQ6	10	The system should be able to add, delete, edit or update the data in forms
REQ7	5	The system must preview all the data in csv format
REQ8	5	The system should be able to import and export files in comma separated values (also known as csv) format
REQ9	5	The system can print and export in pdf/csv searched data in the Compile Report
REQ10	15	The system can search specific data in the database (search as authors, year and tables) with multiple sql queries
REQ11	5	The system can present reports of searched data in the Compile Report
REQ12	3	The system should validate the added forms
REQ13	5	The system should verify the users/admin by User ID on log in
REQ14	10	The system can verify the user's ID in order to allow to add or import any data in the website
REQ15	15	The system allows the administrator to manage the whole system
REQ16	5	The system can create a Compile Report where all the tables can be searched
REQ17	2	The system should be connected to a MySQL database
REQ18	15	The system should use a web server (Apache), MySQL as the relational database management system and PHP as the object-oriented scripting language
REQ19	5	The system should be accessible only on the network of the KIOS Research Centre and will be restricted at public networks
REQ20	5	The system should use Javascript for the back-end functions of the website

Points* = the higher the number of points refer to the higher difficulty of the task

REQ = stands for the **requirements** that the system needs to have in order to fully accomplish the manager's expectations.

So this lead us to the complexity of the project, also known as Technical Complexity Factor, which a table below previews the non-functional Requirements.

Non-Functional Requirement

Non-functional requirements refer to characteristics and constraints with which the system must comply. It concerned with quality aspects of software system and good user experience such as performance, security, availability. Unlike FR, NFR does not usually related to functionality that yields operation results directly. However, NFR do affect the experience or result quality when the user using the system. Thus NFR may affect the overall architecture of a system rather than the individual components."

The supervisor further claimed that NFR might also generate FR given that the example of implementing Security features could introduce new system services. There are several concerns with respect to NFR, especially usability or usability analysis. Users should feel comfortably navigating the system using desktop or laptop devices. The user interface should provide clear indication of navigation path, and the menu, colour and layout should look consistent to the users. Security is also a major consideration when developing the system. Access control should be implemented to prevent destructive actions.

Technical Complexity Factor (TCF) — Nonfunctional Requirements

Technical Factor	Description	Perceived Complexity	Weight	Calculated Factor
T1	Distributed system (running on multiple machines)	3	2	2x3=3
T2	Performance objectives	3	1	1x3=3
T3	End-user efficiency	3	1	1x3=3
T4	Complex internal processing	3	1	1x3=3
T5	Reusable design or code	0	1	1x0=0
T6	Easy to install	3	0.5	0.5x3=1.5
T7	Easy to use	5	0.5	0.5x5=2.5

Technical Factor	Description	Perceived Complexity	Weight	Calculated Factor
T8	Portable	2	2	2x2=4
T9	Easy to change	1	1	1x1=1
T10	Concurrent use (by multiple users)	4	1	1x4=4
T11	Special security features	5	1	1x5=5
T12	Provides direct access for third parties(the system will be used from different organisations)	1	1	1x1=1
T13	Special user training faculties are required	0	1	1x0=0

The TCF is one of the factors applied to the estimated size of the software in order to account for technical considerations of the system. It is determined by assigning a score between 0 (factor is irrelevant) and 5 (factor is essential) to each of the 13 technical factors listed in the table below. This score is then multiplied by the defined weighted value for each factor. The total of all calculated values is the technical factor (TF). The TF is then used to compute the TCF with the following formula =>

$$\text{Column1} = C1 = 33$$

$$C2 = 14$$

$$C3 = TFT = 34$$

Result of Technical Complexity Factor (TCF) => TFC Formula =>

$$TCF = 0,6 + (TF/100) => \mathbf{TF = 34}, \text{ from above Table - Column 3}$$

$$TCF = 0,6 + (34/100)$$

$$TCF = 0,6 + 0,34$$

$$\mathbf{TCF = 0,94}$$

When applying any general cost estimation technique, you have to account for many variables. Every software project is different, and if you don't account for those differences, your estimation will not be reliable.

All factors do not have the same potential impact on a project cost estimate, so each factor has a multiplier, representing the relative weights of the factors.

Context Model

During the initial phases of modelling requirements, developers must first understand the context where the system operates in a specific environment. The context model captures

this perspective to allow decisions on project scope to be made by the stakeholders. The context model also shows system dependencies to its interacting environment. The external dependencies could be another automated system, manual processes, and functional peripheral or actors. They might produce data for system usage or consume the output of the system.

Context Diagram (CD) which depicts overview of system working environment is one of the techniques utilised to model the system context. It shows system interaction with external entities and is used to identify information and control flows among these interactions. The two fundamental components in CD are actor and message. Actor represents the external entities with which the system interacts. It refers to a particular user's role who uses the system to perform task or external systems that are required by system to provide functionalities. Message encapsulates information flow and controls as part of connection between system and actors. Each connection is labelled with information or particular functions that flow between actors and system. These connections provide insight into possible events that the system must response if the message is a particular type of command. For instance, the system will send the information form to database when User submitted form information. A typical message will contain two essential properties: data content and arrival pattern. Data content depicts the information that the message carry while arrival pattern describes the nature of message occurrence and possible events that trigger its occurrence.

The list of actors is identical to the stakeholders described above as they are the primary operators of the system. The Context Diagram also included external actors that the system was interacting with, such as Database. The message flows show typical information used in the environment.

Example description of Form Information Message

Message : Form Information

Data Content Contains form associated with selected table and their authors.

Arrival Pattern Occurs asynchronously when user submit form based on their request.

Context Diagram allows a clear understanding of system dependencies and system scope. Besides, explicit labelling information flow among system and external actors provides initial concept of data structure and process sequence. These ideas will then contribute to design analysis and system architecture.

Use Case Model

Use case modelling is one of the commonly applied modelling techniques in requirement modelling. Its primary use is to capture interactions between users with the system.

Interactions that occur within a system could be user interactions such as input gesture, communication with external systems, or collaboration between components of the system. Knowing users' preferred ways to interact with the system also allows developers to capture precise requirements and build a more usable system.

Use cases are simple descriptions of system features from the point of view of users. Use cases also capture scenarios of what the user could perform with the system and the expected response from system. Nevertheless, a use case is often used to capture functional requirements of the system and generally are inappropriate for non-functional requirement. Use case modelling involves two major artefacts: use case diagram and use case description.

Use case diagram is a simple representation of what functions the system allows actors to perform. It provides a high-level view of the relationship between actors and functionalities. Each use case is represented as an oval shape and each actor is represented as stick figure. An actor could provide input and receive output from associated use cases and these associations are depicted by line. The diagram shows all the actors that interact directly with the system features.

USE CASE DIAGRAM

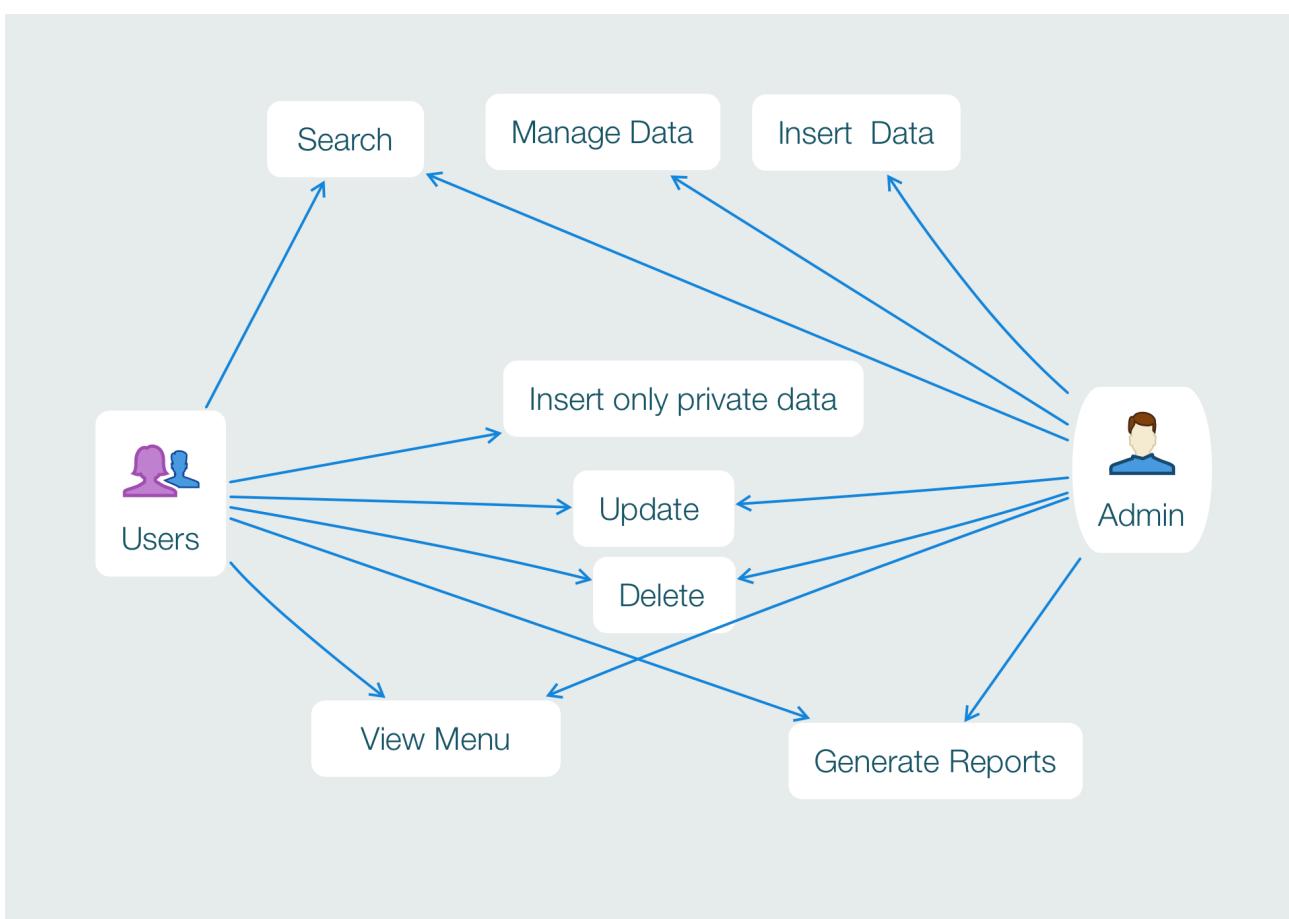


Figure 3 - Use Case Diagram

Description Actors Scenario

User intended to create, update or delete private data.

The use case starts when user enters the menu after the user signs in. The system will present the menu. The user select a table of publication. And then selects what to do with the data inside that table (insert, update, view ,delete). Then the user submit the completed form and the system collects the updated data.Create and Update From will prompt user to input the details. Delete action will selected data from database. After selected action is completed, the user confirms to persist the result.

Use case diagram shows only a small amount of detail about the functionalities and their flows. Therefore, use case descriptions are required to provide in-depth information to understand what is involved. They could be simple textual descriptions, a tabular and structured description, or a sequence diagram. Abstracting the interaction details in the scenario and intentionally avoids premature decisions such as UI element design.

In summary, modelling interactions are fundamental steps to explore essential features required to achieve user objectives. This is particularly useful for grouping related features belonging to an actor role.

Based on the requirements discussed in previous sections, the project needs to formulate a project plan or sets of development activities derived from software engineering methodologies. As discussed earlier , the process agile method will be adopted as the software process model for the project. The important concern of adopting software process model is not strictly follow every principles and steps – but to use it as guiding principles. This section intended to cover some methods that could be used to improve traceability of project activities to requirements. It involves structuring tasks to be performed (i.e. software process model) and consideration of significant milestones.

Small Iteration or Releases

The development activities could be easily organised based on the system modules and features. Each modules depicts a major release of the software features that are then verified against a sub-set of the requirements. It is important to note there are various dependencies between these modules. Some lightweight tasks such as defining interfaces for other modules took precedence to enable development smoothness. At the end of each iteration, there are possible chances to re-evaluate requirements and adjust the plans, once each is: evaluated, tested, and validated against requirements. Each version of release prototype will be traceable to a FR or NFR. This ensures high priority works are focused on first.

2.5 System Architecture

As one of the major concerns in software engineering, the term software architecture could be understood through various definitions. Defined it as:

"The process of defining a structured solution that meets all of the technical and operational requirements, while optimising common quality attributes such as performance, security and manageability".

In addition, defined software architecture by highlighting its important elements, listed as:

- 1) Highest level breakdown of a system into part
- 2) Decisions that are hard to change
- 3) There are multiple architecture in a system
- 4) What is architecturally significant is one that can change over a system's lifetime.

These definitions mean that the software architecture phase represents the important design decision phase required to build a stable and scalable system. This is crucial for a successful system because it helps to address the quality and risk factors aligned to the project. Moreover, design concerns such as selection of algorithm(s), business logics and data structures often overlap with architecture decisions. Thus, the vision of architecture needs to be established to provide direction of development style.

This project is developing a website hence, the following sections will explore several software architecture styles for the website.

Client-Server Architecture

The most common architecture style for a distributed system such as website is the conventional client-server architecture. It divides the system into server components that offer services and client components that provide user interface to consume the services through connected networks. The server could serve request from multiple clients. This architecture style is also known as 2-tier architecture.

Clients are often represented by range of applications with GUI (Graphical User Interface) to capture user inputs and transform them into requests to the server. The server contains data storage to collect, modify and distribute data. Client could be either a thin client if application processing logic is located at server side or a fat client if it is embedded with client application. However, this architecture style lacks scalability because both the client and the server have limited resources. Reusing the web logic for different modules is also increasingly complex as systems expand due to their tight coupling between either data

tier or presentation logic. In addition, it is difficult to maintain because it is hard to distribute new changes to system users.

N-tier and Layered Architectural Style

The N-tier (multi-tier) or most popular 3-tier architecture style addresses conventional 2-tier architecture issues by placing the application logic at an additional tier. Layered architecture shares the common goal with multi-tier architecture style, separation of functionalities into segments to improve scalability and maintainability. N-tier architecture is often used together with layered architecture style and they addressing following concerns:

- i) Presentation logic (user interface)
- ii) Business and website logic (domain process)
- iii) Data accessing logic (database communication)

The presentation tier is at the top most level of the system and exposes visual representation to allow users to interact with the system. It gathers the user intention in the form of commands and inputs and forwards them to the business tier. The business tier is responsible for processing data between presentation and data tier. It contains important application logic performs calculations, evaluation and makes logical decisions. The result of the logic execution will either become responses to the presentation tier request or forwarding to data tier for persistent. At the bottom, the data tier handles create, update, read, and delete operations for the data sources. It contains various information to form appropriate queries for data retrieval. In an OO (Object Oriented) environment, it also in charge of handle mapping between information from data source to domain entity.

Based on the explanation the layered architecture style focusses on grouping related functionality into distinct layers and stacked vertically. Layered architecture model system structures in layers and each layer will have a collection of functions with similar concerns. Each layer can grow independently due to its loose coupling to upper and bottom layer.

In theory, multi-tiered and layered architecture styles have a number of similarities.

However, layered architecture has adopted OO design concepts into its core principles. Each layer promotes abstraction and encapsulation by providing just enough details to the dependant. It should also have high cohesion and clearly defined responsibility for maximised reuse. In contrast, multi-tiered architecture styles are concerned with the deployment of each segments into distributed environment. A 3-tier (layers) software architecture that is commonly adopted by web. As discussed in the previous section, N-tier architecture enables separation of concerns and simultaneously promotes scalability and loose coupling among interaction objects.

The Presentation Layer (PL) is responsible for mapping the information collected from domain entity to the UI element in HTML. In addition to this, it handles certain UI transitions on the client side to allow the application more responsive to user gestures. It is also important to ensure the presentation logic decouples from the business process logic. Separated presentation patterns such as Model-View-Controller (MVC) and Model-View-ViewModel (MVVM) will be utilised to manage communication between the presentation components. These are discussed in details and with the framework used to implement them.

The Business Logic Layer (BLL) is a middleware layer that provides the services of business functionality for PL. It encapsulates different computational logic, such as calculating form total amount or workflow logic, such as updating a menu plan within each domain services. It also contains important business entities that serve as basic data structures to carry and process information. This layer employs an OO design and analysis approach to address the structural and behavioural aspects of the website.

The Data Access Layer (DAL) mainly handles communication between website with the data sources. Its responsibilities include establishing necessary connections to the data source and construct necessary queries to retrieve and persist data. The database context will take care of the life cycle of database connections while entity configuration will adjust database schema on when the entity structures change. The design of this layer is mainly concerned with data structure and data storage methods.

An overview of the physical architecture that will be supported by prototype software. The web, deployed into the web server, will provide centralised access to its services based on the local area network. In addition, a database server will hold the operation data of the system.

With both a vision of software and physical architecture established, the project has set a baseline for further design processes. The next steps will involve producing design models to address the design concepts of these architecture's components.

Chapter 3

Modelling and System Specification

3.1 System Modelling

System modelling is a process to construct abstract representations of a system in several models, with each model encapsulate different views and analysis perspectives towards the system. In fact, it primarily focuses on creating design models. It works closely with requirement modelling by transforming requirement analysis result to design representation for building software. These models encapsulated requirement understanding such as specification of software operational characteristics. Software interface with other system elements and constraints that software must meet. Often, these design models could easily translated from functional and non-functional requirements and vice versa. They are used throughout development process and they are commonly used for:

- A. Facilitating discussions about existing or proposed systems
- B. Documenting an existing system
- C. Acting as a detailed system description, which could be used to generate system implementation.

Design models are often represented by different types of graphical notation with additional labels to describe their meaning. Depending on the development approach, different types of notation could be used to express a particular design. The most notable design notation used at present is UML, a unified modelling language that contains a robust notation for the modelling and development of object-oriented (OO) systems. Considering UML diagrams are de facto artefacts in the analysis and design process. However, this does not cover every representation or variant of design concepts. Additional diagrams such as Class responsibility-collaborator (CRC) Card and Entity Relation Diagram (ERD) will be adopted to express uncovered design concepts. To follow the model with a purpose principle, models that are meaningful to the development process. While various types of models existed, modelling perspectives that are critical are:

- a. Behaviour model, describes dynamic behaviour of a system and its response to events;
- b. Structural model, describes organisation of system structure and how its information is been processed
- c. Data model, describes information that will be persisted and their relationships.

Each model could employ several possible modelling techniques and artefacts to represent its perspectives. The project goal does not exhaustively produce every artefacts of these models. It is rather to investigate how some modelling techniques could be applied to achieve the design objectives. The models also may not include every fine-grained level of details, yet these initial models will continue grow as developer refactors the design of the system.

Structural Model

A complex system consists of several sub-components and possibly some external systems. Structural model displays the organisation of these components and their relationships. It embodies important consideration about entities that will operate within the system. In addition, structural model is to create a vocabulary that can be used by the analyst and the users. Things, ideas, or concepts discovered in the problem domain are represented as given object types in structural model, including relationships among such objects. In fact, this process is also known as domain modelling in OO development. Class responsibility-collaborator (CRC) cards and Class Diagram are the two focuses of the structural modelling techniques.

CRC card is a collection of standard index cards that have three major sections as specified below:

- 1) Class, represents a template used to create specific collection of similar objects;
- 2) Responsibility or operation
- 3) Collaborators, represents other interacting classes that are required for obtaining necessary information to fulfil responsibility.

3.2 UML State Machine Diagrams

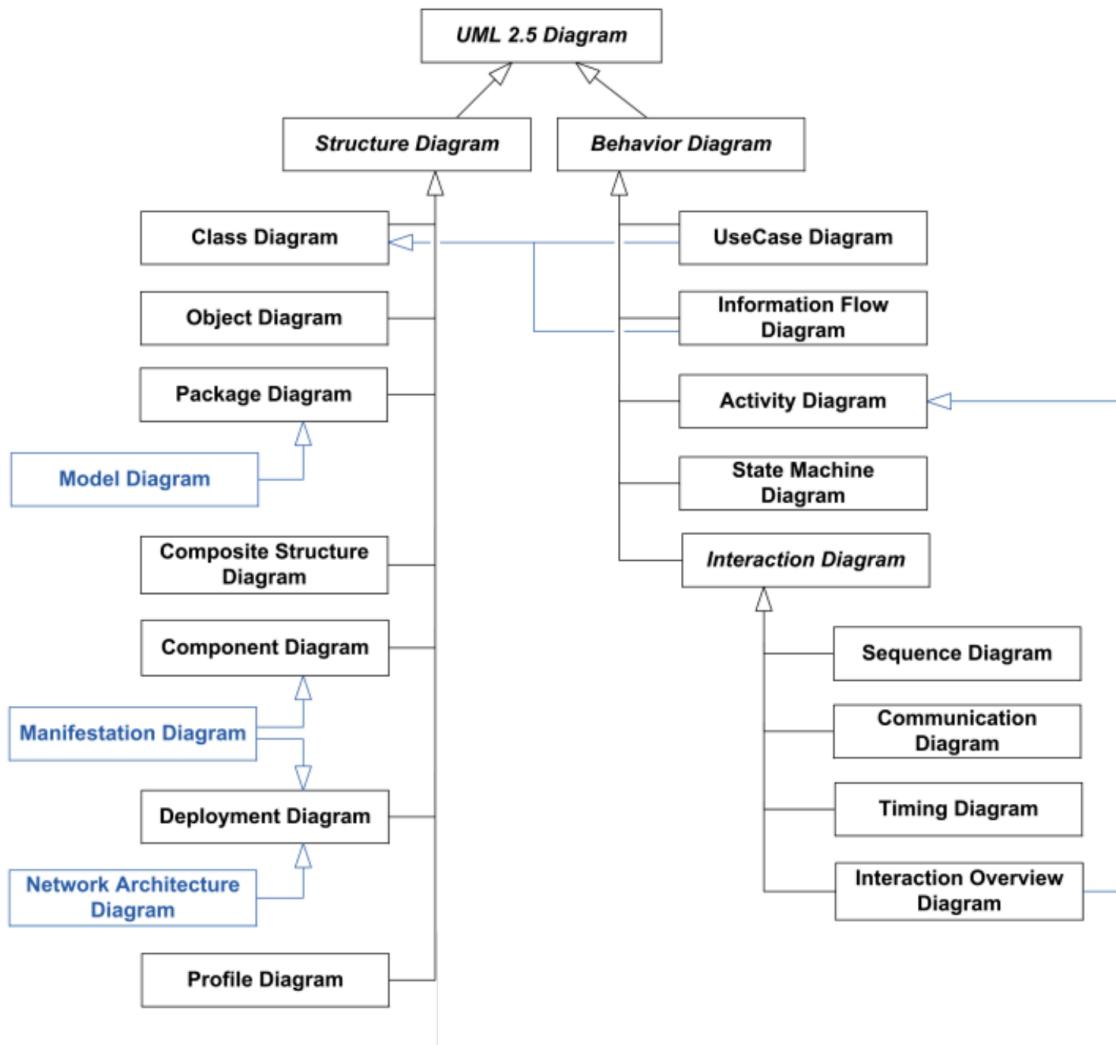


Figure 4 - UML Diagram

Class name will present at the top, while responsibilities and collaborators will present at left and right panel respectively below it. CRC modelling is good option to explore structure and content of a particular class. Experimentally walking through use case with these cards could uncover design flaws such as missing properties, coupling and cohesion. The class knows some information and they are often translated into property, while other responsibilities will become operations.

The established CRC model will eventually be translated into Class Diagram. The Class Diagram is a model that shows attributes and operations of system classes, as well as relationships among them. Class Diagram can have two levels of detail:

- i) analysis class diagram

ii) design class diagram

The Analysis Class Diagram is used to understand the relationship between domain objects while the Design Class Diagram contains in-depth descriptions of object's attribute and operations. In fact, Analysis Class Diagram is also known as domain modelling in OO development. It illustrated the possible elements that will be implemented as actual class.

Behaviour Model

Behaviour model aims to model the dynamic behaviour during the execution of the system. It can be used to describe a use case at a specific time and event. Behaviour model is particular useful to model a possible business process. The business process can be expressed in a series of continuous interacting objects in the system. Suggested that there are two types of behaviour modelling.

The table shows a comparison of two types of approaches in behaviour modelling. It describes the purpose of modelling when they should be used and the tools that could help during the modelling phase. As a system to support business operations, expecting various explicit inputs from end users. For instance, when a User submits a form, the type of table is specified. Hence, data driven modelling is adopted because it fits the data processing nature. In addition, since the modelling process focuses on OO design, Sequence Diagram is used to model behaviour as opposed that is mostly used in a structured analysis design.

Data driven modelling

Shows the sequence of actions involved in processing input data and generating an associated output. Useful when used to show entire sequence of actions that take place from an input being processed to the corresponding output Data-flow diagrams (DFD), and sequence diagrams.

Event Driven Model

Shows how a system responds to external and internal events.

System has a finite number of states and that events may cause a transition from one state to another. Sequence Diagram is one of the UML diagram to objects, including actors. It demonstrates a sequence of interaction activities during a system flow. The interaction activities could be a reflection of an explicit sequence of messages that have passed between objects [25]. All the objects will be arranged in a parallel line and a vertical dotted line indicates its active timeline. It is specified that not every detailed should be

included in a Sequence Diagram unless it is used for code generation. The reason being it may lead to a lot of premature implementation design. Eventually, it could easily fall into entropy of big design upfront.

Furthermore a Sequence Diagram to capture behaviour to submit a new order. It consists of high-level view abstraction on the potential classes and messages exchanges required to be carried out to submit an order use case scenario. Messages flows will inspire operations that need to be implemented for the potential classes.

Data Model

Data model is concerned with exploring data-oriented structures [59]. It aims to define the structure of data objects, their relationships and related information that describe the objects and relationships. Data model is important because every application, at a certain point, will need to persist its data to certain type of storage. The structure of data, if not carefully designed, will affect data retrieval performance. The data model design is closely related with the decisions of data structures. Before this process can even begin, the type of data and model behaviour and interaction between storage strategy needs to be investigated and carefully selected to accompany the data processing need within the system.

The Context Diagram lets the designer (and/or the developer), realise a view of input and output data. The views are further developed into more concrete ideas regarding inner content of data, through structural models. This section develops the concepts related to the explicit data structure concern in this project.

The first issue, if data structure is concerned and related to this project, that needs to be addressed is: persistent or non-persistent. Once the decision of persistent or non-persistent has been made, the next issue is whether this data is going to be stored externally or internally to the application. This implies an external data store that will need to be selected, designed and implemented. In fact, it involves designing different types of data structure. If different types of data structures need to be designed, we need to think about different type of physical files that need to be processed. Hence, the next decision would be the type of files: whether they are simple or complex. If the file is going to be processed implicitly, it could be a text file, or comma-separated values (CSV) file. However, if the file required more processing and involves complex read-write operation, it will be more appropriate to select a formal mark-up file as the data store. This could be a data structure of relational model or hierarchy model (e.g. Extensible Mark-up Language).

3.3 Relational Database Management System (RDBMS)

RDBMS is data-processing software that employs relational model as its fundamental data structures. RDBMS describes the relational model as following in the relational model, all data is logically structured within relations (tables). Each relation has a name and is made up of named attributes (columns) of data. Each row contains one value per attribute.

It presents the data in tabular form identical to spreadsheet format. The standard language used for data manipulation in RDBMS is Structured Query Language (SQL). SQL consists of two major components: Data Definition Language (DDL) for defining the data structure and controlling access to data and Data Manipulation Language (DML) for retrieving and updating data. SQL can be very powerful depending on the usage. Some applications leverage its capability to transfer part of the system computation logic to RDBMS through writing stored procedures with SQL.

RDBMS has become the dominant data storage methods for most software systems today, particularly websites. There are many existing RDBMS solutions that all share similar sets of essential data processing functions varying slightly in the provided features. There are several mature commercial solutions such as Microsoft SQL Server [62] and Oracle Database. Conversely, there are also free open source solutions such as MySQL. These have been widely adopted by industry, particularly in small and medium business. The advantages of RDBMS are:

- A. Support for controlling concurrency and transactional access;
- B. Support for security management;
- C. Minimal data redundancy
- D. Simple user interface to manage data schema
- E. Ensuring data integrity through constraints
- F. Fast data retrieval through query optimisation

Extensive Mark-up Language (XML)

XML is a meta-language that allows designers to define their own customised tags in a document. As a type of semi-structured data, XML is designed to be self-descriptive, readable by both humans and machines. It has a loose restriction of schema, thus allowing it to handle data structure that changes rapidly and unpredictably. This is especially true for information on the Web, where it requires certain degree of flexibility to accommodate ever-changing HTML design. The software industry today uses XML as the de facto standard for data communication. It has evolved to be the primary medium of data exchange with external systems and among businesses.

Many technologies have built upon XML by a predefined structured format for XML with XML Schemas. These schemas lead to standardisation of XML format when adopted widely by the industry. SOAP and RSS are some of examples of spin-off technologies based on XML. XML also has become popular thanks to a wide range of query languages available.

XML could be considered as the data storage if following advantages can be utilised:

- A. Ability to deal with frequent and unpredictable schema changes;
- B. Modelling hierarchy data structure effectively
- C. Minimum data conversion if data is used directly from source; and
- D. Support for multiple platforms.

Storage Method Chosen

The storage method chosen for both persistent files (Application Setting and Business Entity) is the RDBMS. The main rationale was the Business Entity, primary data structure, required extensive cross-referencing. For instance, an order needs to know which recipes been selected and the recipes need to know what are the materials that need to be consumed. Putting this data into hierarchical models will result into redundant data everywhere. RDBMS is also has strong security measures (access control) and they are important for Applications/Websites Setting, which contains critical data that would affect entire system behaviours.

In addition to that, another key factor that leads to this decision is the existence of Object Relational Mapping (ORM) solutions. ORM solution mainly help in converting the relational model into interconnected object graph, coined as the “Object-relational Impedance Mismatch” problem. This significantly reduces the complexity of retrieval relational data into OO environment because complex SQL queries could be simplified into normal OO method calls.

Conversely, using XML as data storage could be relatively complex and verbose. The processes to retrieve the data in documents, map them to object and primitive data type in programming language, is difficult. Because the data type constraint is not enforced within XML, the designer will need to handle various kinds of data processing issues such as null value and format mismatch.

Entity Relationship Diagram

Entity Relationship Diagram (ERD) is widely used data model to represent relational data model of database. It is a picture which shows the information that is created, stored, and used by a business system. There are three major concepts in ERD. First, each data object in ERD is a named entity, often mapped to a table in RDBMS. Second, information with an entity is represented by a set of attribute, which capture the data segment (table title) of a data object (author). Finally, association among entities, also known as relationships, depicts high level business rules of a system. This model is reflecting actual implementation in the database for data persistence.

Chapter 4

Data Representation

4.1 Description of Database

Database is a collection of logically related records (or data). It may consist of enormous number of data. Hence it is really important to manage and organise data within the database. Database Management Systems (DBMSs) are software applications that use to interact with database and the user by giving capability of managing data. It allows tagging, retrieving and manipulating data efficiently and quickly while ensuring the security and unauthorised access. SQL Server is one of the well-known DBMS used today. Web-based System uses a database for storing and managing records of loans and reservations, equipment details and user details. It uses basic database interactions such as insert, select, delete and update in different scenarios (for example when registering equipment or user, displaying equipment details, modifying and deleting equipment or user details, inserting loans and reservations).

4.2 Apache Server

The Apache HTTP Server, informally called Apache, is the world's most popular web server software that in 2009 it became the first web server software to serve more than 100 million websites. The Apache development began in early 1995 and originally based on the NCSA HTTPd server. Apache is developed and maintained by an open community of developers under the patronage of the Apache Software Foundation. Mostly used on a Unix-like system, the software is also available for a vast variety of operating systems, including Microsoft Windows, Open VMS, eComStation, NetWare and TPF.

Apache is open source software, as on November 2015, it was estimated to serve 50% of all active websites and 37% of the top servers across all domains.

4.3 MySQL System

SQL stands for Structured Query Language. MySQL is an open source Relational Database Management System (RDBMS). It is a popular database for use in web applications, and is a central part of the greatly used LAMP (Linux, Apache, MySQL, Perl/PHP/Python) open-source web application software stack.

MySQL is used by many applications like, WordPress, Joomla, TYPO3, Drupal, MyBB, phpBB, MODX and other software. Numerous large scale websites including Google, YouTube, Facebook, Twitter, and Flickr are also using MySQL.

On all platforms excluding Windows, MySQL sends with no GUI (Graphical User Interface) to administer MySQL databases or managing the data held within the databases. Users may install MySQL Workbench by downloading separately or simply may use the command line tools. Numbers of third party GUI tools are also available.

Swedish company has created MySQL which is written in C and C++. The first version of MySQL revealed on 23 May 1995. It has various versions. The general accessibility of MySQL 5.7 was broadcast in Oct 2015, and the version which is used in my project is 5.6.17.

4.4 List of Database Tables

Database Tables

JOURNALS TABLE

Author(s)
Paper title
Journal name
Volume
Number
Pages
Date

BOOKS TABLE

Author(s)
Title
Publisher
Place (Optional)
Pages (Optional)
Year

CONFERENCE TABLE

Author(s)
Title
Conference
Pages
Date

BOOKS CHAPTERS TABLE

Author(s)
Title
Pages (Optional)
Editors
Publisher
Place (Optional)
Year

OTHER TABLE

Author(s)
Title
Appears in: (volume ,number, pages)
Pages
Date

PATENTS TABLE

Author(s)
Title
U.S. Patent
Date Issued

Chapter 5

Design and User Interface

5.1 Design Principles

System design involves designing a system based on user requirements, and testing the feasibility of the project. In general, the steps for a prototype system design include: database design and populating, system architecture design, user interface design and functionality development

Analysis model

Often a design element corresponds to many requirements, therefore, we must know how the design model satisfies all the requirements represented by the analysis model.

Programming paradigm

A programming paradigm describes the structure of the software system. Depending on the nature and type of application, different programming paradigms such as procedure oriented, object-oriented, and prototyping paradigms can be used. The paradigm should be chosen keeping constraints in mind such as time, availability of resources and nature of user's requirements.

Uniform and integrated

Software design is considered uniform and integrated, if the interfaces are properly defined among the design components. For this, rules, format, and styles are established before the design team starts designing the software

Flexible

Design should be flexible enough to adapt changes easily. To achieve the flexibility, the basic design concepts such as abstraction, refinement, and modularity should be applied effectively.

Minimal conceptual (semantic) errors

The design team must ensure that major conceptual errors of design such as ambiguous and inconsistency are addressed in advance before dealing with the syntactical errors present in the design model

Degrade gently

Software should be designed to handle unusual changes and circumstances, and if the need arises for termination, it must do so in a proper manner so that functionality of the software is not affected

Correspondence between the software and real-world problem

The software design should be structured in such a way that it always relates with the real-world problem

Software reuse

Software engineers believe on the phrase: 'do not reinvent the wheel'. Therefore, software components should be designed in such a way that they can be effectively reused to increase the productivity.

Designing for testability

A common practice that has been followed is to keep the testing phase separate from the design and implementation phases. That is, first the software is developed (designed and implemented) and then handed over to the testers who subsequently determine whether the software is fit for distribution and subsequent use by the customer. However, it has become apparent that the process of separating testing is seriously flawed, as if any type of design or implementation errors are found after implementation, then the entire or a substantial part of the software requires to be redone. Thus, the test engineers should be involved from the initial stages. For example, they should be involved with analysts to prepare tests for determining whether the user requirements are being met.

Prototyping

Prototyping should be used when the requirements are not completely defined in the beginning. The user interacts with the developer to expand and refine the requirements as the development proceeds. Using prototyping, a quick 'mock-up' of the system can be developed. This mock-up can be used as an effective means to give the users a feel of what the system will look like and demonstrate functions that will be included in the developed system. Prototyping also helps in reducing risks of designing software that is not in accordance with the customer's requirements.

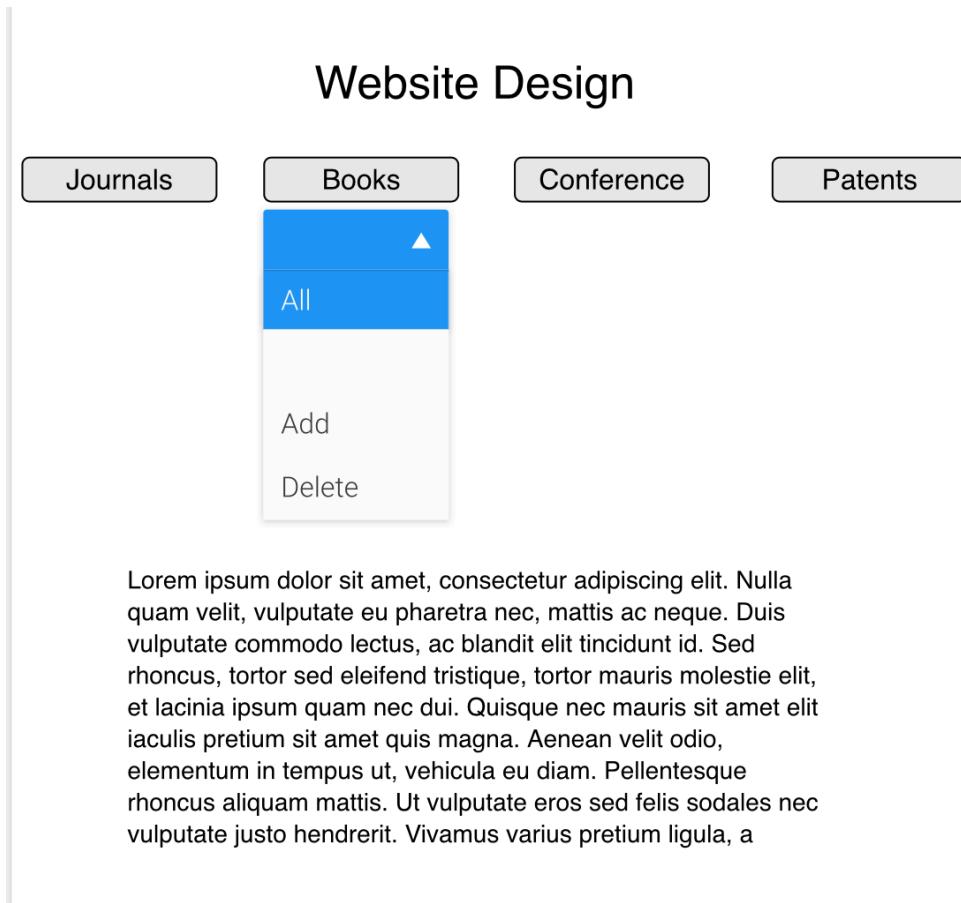
Note that design principles are often constrained by the existing hardware configuration, the implementation language, the existing file and data structures, and the existing organisational practices. Also, the evolution of each software design should be meticulously designed for future evaluations, references and maintenance.

5.2 System Design

How important is user-interface design? The design of a user interface, in this case the design of a form, can clearly be critical. You must first understand the basics of user interface design, there may very well differentiate the outcome and the feel of the user

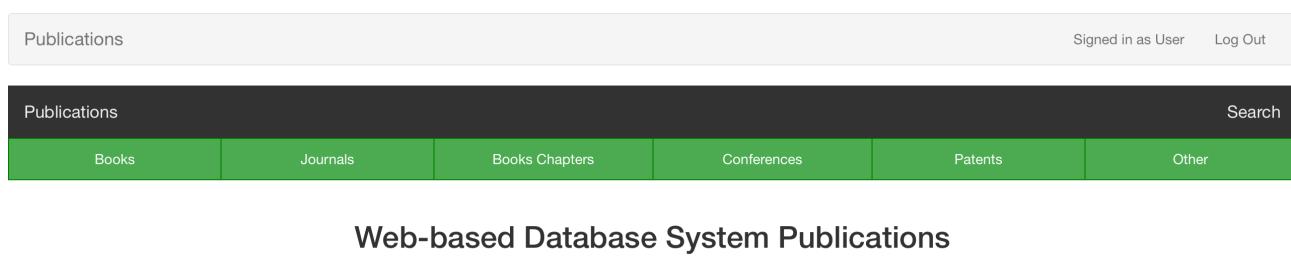
experience. Granted, this is an extreme case, but the user interfaces that you design can have a significant impact on the success of an organisation, particularly in the competitive environment for systems on the Internet. A living example is Apple, which is known for its beautiful User Interfaces and Graphical User Interface in software. Below I will describe the techniques that helped me improve and also design my Web-based user interface designs, and avoiding a debacle of butterfly-ballot proportions.

Figure 5- Initial Home Page Design



This screen is easier to read, but it still needs improvement.

Figure 6 - Final Home Page Design



Consider the design of the data entry depicted in the initial page. HTML pages accomplish the same task, enabling definition of basic employee information, although by making a few simple changes the second version of the home page is easier to understand and hence to use.

The most noticeable change between the two designs is the use of colour. If you use colour in your application or website, you need to ensure that your screens are still readable by following the contrast rule: Use dark text on light backgrounds and light text on dark backgrounds. Many organisations make the mistake of forcing their Web designers to use their corporate colours as the basis for their user interface colour scheme —you only have to surf the Web for a few minutes to discover this. The colours of your corporate logo may look great in a magazine ad but will often prove to be horrendous for your corporate Web site. eBay has a very colourful logo, yet the company uses black text on a white background for the majority of its Web site, the same colour scheme used by Amazon.com, Chapters.ca and the Barnes & Noble e-commerce sites. Black text on a white background provides the greatest range of contrast available.

A small but significant percentage of the population is either fully or partially colour blind —for many people, the red text looks black, making it impossible for them to distinguish, thus this lead me to avoid red colour completely. I also moved the required fields to the top of the page, grouping them together to hopefully increase the chance that they are filled in the first time. The problem with this approach, however, is that the fields may not be in a natural order, something that user-interface walkthroughs with my users would reveal.

I decided to choose GREEN colour for the buttons and the backgrounds and the pages because it is a beautiful colour, easy on the eyes, therefore you can use the system for hours and not feel your eyes tired. Final page design depicts improvements to the wording of several of the field labels. The text you display on your screens is a primary source of information for your users. If your text is worded poorly, then your interface will be that much more difficult to use. Using full words and sentences, as opposed to abbreviations such as "dt" for date and "Addr" for address, makes your text easier to understand because your users don't need to perform any translations. Furthermore, each label is capitalised, a minor but important improvement as it promotes consistency, which in turn makes your user interface easier to understand and to learn. Final design also drops the all white in front of the page , because too much white might get tiring after a while. Similarly, any messages displayed to your users should be worded positively to imply that the user is in control and provide him with insight into how to use the web properly. Error message is more appealing: "You have input the wrong information". Furthermore, messages are worded consistently and displayed in a consistent place on the screen. Although the instructions "The journal name must be all characters" and "A start date should be input" are worded well when considered separately, together they are inconsistent. A better and more consistent wording of the second message would be "The start date must be input." Although neither of the figures shows a message, wording is an issue for any results returned by your data validation routine for this HTML form—either a descriptive error message or a confirmation that the system has received the information. This is an easy way to make your system easier to understand, while improving the quality of the data being input. I removed the use of red text for required

fields because the use of asterisks is sufficient for this purpose. I also removed the box around the address fields: Had the page been more complicated, I likely would have left it in.

The most important thing you can possibly do is ensure your user interface works consistently. If you can click on items in one list and something happens, then the same sort of thing should happen when you click on items in any other list. Buttons in consistent places on all your pages, I've used the same wording in labels and messages, and I've used a consistent colour scheme. Consistency in your user interface enables your users to build an accurate mental model of the way it works, and accurate mental models lead to lower training and support costs.

Navigation between pages, screens and report items is important. If it's difficult to get from one page to another, then your users will quickly become frustrated and give up. When the flow between HTML pages matches the flow of the work the user is trying to accomplish, then your website will make sense to your users. Because different users work in different ways, your system needs to be flexible enough to support their various approaches. User interface-flow diagrams should be developed to further your understanding of the logic flow of your user interface, enabling you to address areas that need improvement.

Similarly, navigation within a page is important. In Western societies, people read left to right and top to bottom; therefore, if this is your target audience, your user interfaces should be designed to reflect this reading pattern. In short, navigation between widgets on your screens or pages should be organized in a manner that your users will find familiar.

When designing systems, expect your users to make mistakes. How many times have you accidentally deleted some text in one of your files or the file itself? Were you able to recover from these mistakes or were you forced to redo hours, or even days, of work? To err is human, so you should design your user interface to recover from mistakes made by your users.

An important factor in user interface design is considering access for the disabled, particularly people who have impaired vision. At a minimum, you should always have text representations for any graphics displayed on your Web pages. Chaos theory tells us that the flapping of a butterfly's wings in Thailand can cause a hurricane in the Caribbean. User-interface design theory now tells us that the poor design of a butterfly ballot in Florida can embarrass the U.S. throughout the world. If there is one message you should take away from this column, it's that you don't need to make your user interface flashy to make it effective. The page design must be fancy, but also should get the job done. In fact, the most successful e-commerce sites on the Web focus on functionality over flash. Techniques such as the appropriate use of colour, field alignment, text wording, consistent field design and the appropriate use of user interface widgets go a long way to the development of good user interfaces. I took the time to learn the basics of user interface design and I hope the users will thank me for it.

5.3 User Interface Design

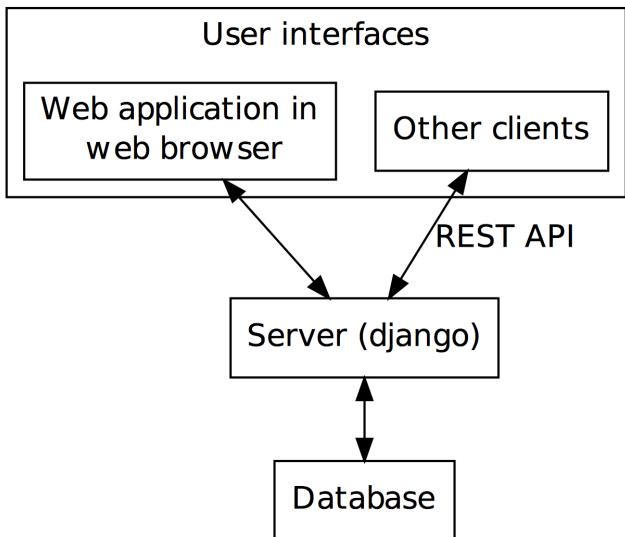


Figure 7 - User Interfaces Diagram

UI Design remains one of the main topics of the project. The UI concerns of this project are mainly designing UI that comfortably map user task to the interaction action, and more mobile friendly website design. The following section will introduce a design approaches called Task-based UI design. In the next diagram it shows the connection between the web browser the server an the database of the system.

Introducing Task-Based UI Design

UI design has been increasing challenging for developers due to the rise of complexity and need for consistency. One possible factor that has contributed complexity is the users' mental model mismatch with the conceptual design of UI designers. Users tend to focus on the task in hand rather than struggling to learn the features and procedures necessary to activate services. Besides, it is also challenging to maintain consistency when more features are added to the UI. Users potentially distracted from finding desired command if many possible selections are provided. Thus, new kind of user interactions that are self-explanatory and has a minimum learning curve is required.

Most of the current UI design focuses on the viewpoint of domain objects, which hold underlying data that will be presented in the UI. Users have to learn the navigational system, the naming of interaction objects, and follow a series of navigation on the domain-oriented menu to achieve particular goal. The original intention of users has switched from simply submitting a form to interacting with several domain objects. There are additional layers of transformation from user intention to following the structure of system objects. A Task-based UI employs a different approach by presenting available commands that meet user goals. Using the previous example, taking submit button function will present a more task-oriented command such as "take form information" and arrange the interface to allow to send the data in the database.

Task-Based UI design

Task-based UIs provide better performance in achieving user goals. It is also less cognitively demanding as it allows direct mapping of interface commands to user tasks. Task analysis is an essential technique to model user tasks and system behaviour in

Task-based UI design. It helps developers to understand possible user actions and subsequently transfer them into UI designs.

The next approach to this problem is Responsive Web Design. Responsive Web Design has gained momentum since the introduction of HTML5. With the help of the Cascading Style Sheet (CSS), a HTML formatting framework, it is possible to design a web UI that will adapt to various screen resolution sizes. It scales down or hides certain UI contents to

improve presentation on different desktops. While this approach reduces significant efforts of maintaining redundant sites, it also increases the burden of web browser. As specified less visible content does not imply less content has been downloaded.



Figure 8 - Scalable UI Design

Recent website developments lean towards the responsive design approach. This mainly because it has received increasing support from JavaScript web frameworks such as Bootstrap, jQuery UI and Less Framework. These frameworks adopt a fluent grid concept to layout contents on different screen resolutions. In fact, this project favours this approach because it requires smaller overheads to maintain separate sites and possible to achieve maximum reusability.

Menu Design

After taking account the previous design approach we can clearly see the menu of the website. Which is a scalable design for different screen resolutions.

Publications	Signed in as Christos	Log Out			
Publications	Search				
Books	Journals	Books Chapters	Conferences	Patents	Other

Web-based Database System Publications

This is a website where the publications are going to be published. Each user will have different privileges and can see his own publications or search for his publications. Also for administrator privileges there will be an option to view and search and publish all publications. Additionally after the user enters each of those options, there will be a new page where there is first going to be a text box asking for the name of the publication . If the publication already exists in the system the rest of the fields are going to be added automatically. Note: During the import settings there must be an option where the user have the option to replace completely overwrite to the latest version of the publication or import a new one. All of these publication are going to be added dynamically and you should be able to preview them at tables. To the system there is going to be only one admin (secretary) but there are going to be several users who can be added dynamically after signing up.

Table menu for Books

Books Menu

[Add](#)

[Delete](#)

[Search - Update](#)

[Preview - Import - Export](#)

[Back To Menu](#)

Form Design for Books

Books Form

*For the required fields

Author(s) *

Title *

Publisher *

Places *

Pages Length *

Years *

[Submit](#)

[Back](#)

Chapter 6

Programming Overview

6.1 PHP

It stands for PHP: Hypertext Preprocessor but, originally stood for Personal Home Page. Is a server side scripting language that designed for web development, as well as used for general purpose language. It was created in 1994 by Rasmus Lerdorf, in the present time the reference execution of PHP is produced by the PHP group.

In January 2013, PHP was installed on more than 240 million websites, and 2.1 million web servers. The PHP code can be combined with several web frameworks and templating engines or simply it can be mixed with HTML code.

The PHP code is generally processed by a PHP interpreter, which is commonly executed as native module of web server or a Common Gateway Interface (CGI) executable. After interpretation and execution of the PHP code, the results will be sent by web server to its client.

There are many versions of the PHP, and the version, I have used for my system is the **PHP version 5.5**. But a wise question would be the **reasons to choose PHP**.

Well, I chose PHP because it has interactive features and allows you to interact with your visitor in ways HTML alone could not. So for example simple things like email forms or like shopping carts that save your previous orders and suggest other products. It supports social things like interactive forums and private messaging systems.

Additionally PHP is easy to learn and a lot easier to get started. By learning just a few simple functions, I was able to do a lot of things with the website.

Furthermore works great with html, because I have already built some websites in the past and I am familiar with HTML, making the step to PHP easier. PHP and HTML are interchangeable within the page. PHP might add some new features to my website, but the basic appearance is still created with HTML. PHP is highly learnable. PHP is used for web oriented scenarios that need to be developed quickly, and need to perform really fast. In addition PHP sports an enormous community. Majorities are new to PHP but once you learn the basic, you can reach dedicated professional like Phil Sturgeon, Josh Lockhart. There are excellent resources like Site Point's forums, Stack Overflow and PTRW, problems encountered in PHP are easily solved, and progress is usually both rapid

and highly educational. Significantly important is the fact that the number of open source PHP projects as well as books and courses someone can learn are plenty and with some

Updated Journals Form

Displaying Changed Record Sent

Name: Networks
Author: Polycarpou
Journal: Networking
Volume: 20
Number: 11
Pages: 19-150
Date: 2009-10-10

[Create New Journal](#)
[Menu](#)
[Update Journal](#)

patience and guidance you could become proficient in this language in a relatively short period of time. PHP significantly has matured since a period of time people tried to bash it, and is now inheriting more and more modern features from other languages. Prejudice is prevalent and an individual should learn to recognise it and appreciate it. "It's not the tool, it's how you use it". PHP is now used on more than 80% of the world web servers. It has been helped in no small part by Word press.

Open Source

PHP is open source, primarily aimed at web development and applicable to similar projects. The code of PHP can be placed in any file which is interpreted by the PHP engine, typically once with a .php extension. You just enter the URL which maps to that file in your browser and you're done. Admittedly, the code will only run via a web server with PHP installed. PHP has a built-in server. Although is even easier upload your file to almost any web host. PHP is conceptually simpler than any other language and is

currently winning. Knowing a few PHP statements can write something useful. It has more software dependencies, but PHP concepts are less daunting to new developers.

Deep code base

The Web is filled with PHP code. The most popular platforms for building websites (WordPress, Drupal, Joomla) are written in PHP. Not only are the platforms open source, but so are most of their plug-ins. There's PHP code everywhere, and it's waiting for you to download it, modify it, and use it for your needs.

Simplicity

There's not much to PHP: a few variables and basic functions for juggling strings and numbers. It's a thin layer that doesn't do much except move the data from port 80 to the database and back. That's what it's supposed to do. A modern database is a magical tool, and it makes sense to leave the heavy lifting to it. PHP is the right amount of complexity for a job that's not supposed to be complex.

No client app needed

All of the talk about using the same language in the browser and on the server is nice, but what if you don't need to use any language on the browser? What if you ship the data in HTML form? The browser pops it up, and there are no headaches or glitches caused by misfiring JavaScript threads that try to create a page on the browser from two dozen Web service calls. Pure HTML works more often than anything else, and PHP is optimized to create that. Why bother with JavaScript on the browser? Build up everything on the server and avoid overloading that little browser on the little phone.

SQL

PHP was built to co-exist with MySQL and its many variants, like MariaDB. If MySQL isn't exactly right, there are other great SQL databases from Oracle and Microsoft. Your code can switch with a few changes to your queries. The vast SQL world doesn't end at its borders. Some of the most stable, well-developed code will interface with an SQL database, meaning all that power can also be easily integrated into a PHP project. It may not be one perfect, happy family, but it's a big one.

Speed of coding

For most developers, writing PHP for Web apps feels faster: no compilers, no deployment, no JAR files or preprocessors -- just your favourite editor and some PHP files in a directory. Your mileage will vary, but when it comes to banging a project together quickly, PHP is a good tool to use. PHP is a well tested technology.

Easier to mix Code

PHP makes it easy for the developers to mix code with content. One can simply open PHP tags and write codes without any need of templates or other files. It is designed to give the programming power at the fingertips of a developer.

Deep History

PHP code has a deep history in web development. All the major CMS platforms, like, WordPress, Joomla and Drupal, including their plugins are all written in PHP, making it easy to use and modify it according to one's needs.

Simple Language

PHP is a simple and thin layer language with less variables and certain elementary functions. It's a perfect option for a job that doesn't require much complexity.

Supports MySQL

PHP supports MySQL and its other family members like MariaDB. By interfacing most stable codes with SQL database, PHP empowers its projects with extreme flexibility. When it comes to delivering a project quickly, PHP is usually the developer's first choice. Without using any JAR files and compilers, it enables a developer to create a web application with simply an editor and PHP files.

Widely Used

I have learned PHP because it is the most widely used, and was designed for back end server programming.

Why not Node.js

Node.js is far more recent, less commonly used, is bases on JavaScript which is quirky and unusual in its design. It is not really designed to be good for server side programming, but is a compromise for programmers who are more familiar with client side methodology, who wanted to keep the same language features when doing server side. And I think that is a bad ideal, because I don't think JavaScript is a good language at all, for client or server side. Although, Node is providing the best possible solution, but there are always two sides of the mirror. Wisely think when making a Node application because Node.js is not suitable for processor intensive tasks and any CPU-intensive code makes it really non-scalable. Majority of the web's hosting providers are able to provide PHP hosting. With PHP, you can simply install WAMP, LAMP or MAMP and off you go. For deploying code into a web host, you just drop your files there and you're done. As PHP code runs each in its own process and the web server sits in front and directs this, if one of the requests

causes an error it will only affect that specific request. In PHP each process is only alive for the duration of the request. This means you don't need to worry as much about resource allocation and memory. PHP's standard library is much bigger than Node's.

6.2 HTML

HTML stands for Hypertext Markup Language and CSS stands for Cascading Style Sheets are the crucial technologies for creating web pages. HTML supplies the structure of the page, and CSS the layout, for diversity of devices. Together with scripting and graphics, HTML and CSS are the fundamental of building Web Applications and Web pages.

HTML provides designers and developers the following facilities,

- a. To design forms for directing transactions with remote services, for use in making forms, searching for information, and others
- b. Retrieving online information through hypertext links.
- c. To include video and sound clips, spread sheets, and other applications straight in their documents
- d. Designer can publish online documents with text, headings, tables, photos and others.
- e. CSS describes the Web pages presentation, involving layout, colours, and fonts. It enables the designer to adjust the presentation to various types of devices, like a small screens, large screens, or printers.
- f. CSS is separate from HTML, and their separation makes it easy to preserve and maintain sites, share style sheets across pages, and accommodate pages to various environments

6.3 PHPMyAdmin

It is an open source tool and also, it is free written in PHP, XHTML, CSS, and JavaScript planned to manage the administration of MySQL by using of a web. It is able to perform various missions like creating, modifying databases, tables, fields, executing SQL statements or managing and supervise users.

PhpMyAdmin is being translated into 72 languages in order to make the usage easy to a wide domain of people and it supports both LTR and RTL languages.

Following is some features of the phpMyAdmin:

- A. It is web interface
- B. It administers multiple servers
- C. It is able to create PDF graphics of the database layout
- D. Importing data from SQL and CSV
- E. Export data to different formats such as SQL, PDF, CSV, XML and others
- F. It works with various Operating Systems

6.4 Javascript

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to JavaScript, possibly because of the excitement being generated by Java. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

Client-side JavaScript

Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.

It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.

The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.

The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.

JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript

The merits of using JavaScript are:

1. Less server interaction – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.

- 2.Immediate feedback to the visitors – They don't have to wait for a page reload to see if they have forgotten to enter something.
- 3.Increased interactivity – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- 4.Richer interfaces – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

6.5 Solutions to Selected Problems

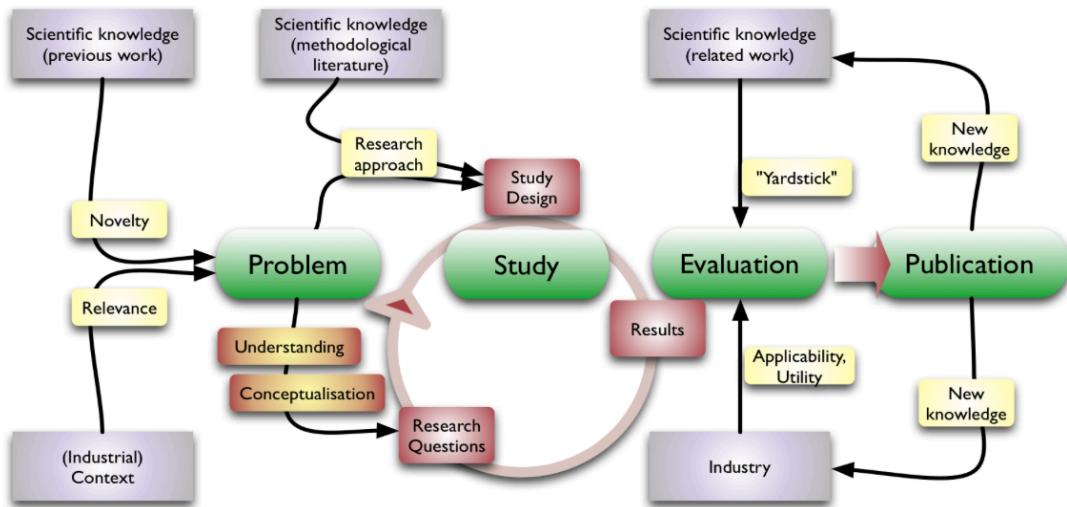


Figure 9 - Problem Research Diagram

Undoubtedly, throughout the whole time I have encountered many problems while I was trying to build the website. Some of them were easy to solve and some of them took me weeks, even months to figure out but thankfully I was able to resolve all of them. I followed the above figure to come through all of the problems I faced during the implementation of my project.

First of all I would like to describe the problem I had with the date format. In my region the common date format is Day/Month/Year (DD/MM/YYYY) , and my advisor asked me to implement this date format inside all the date input forms. But inside the MySQL database the date MUST be in YYYY - MM - DD.

My solution to the problem was firstly create a function that reads the date format in DD-MM- YYYY and converts it to the wanted format of DD/MM/YYYY. But after software testing I realised that the dates were not inserted as described inside the database due to the slash '/' day will mix up with months, and also the validation of the form had some errors while trying to convert in the correct format.

This made me realise that the appropriate way to do it , is to go with the flow. Instead of trying to make complicated functions that might work and might put some wrong day or month inside the database , I figure out that the format should be the format of the database inside the input of the form , thus sending the date inside the database would not cause any problem.

Furthermore , creating the search box in order to search the whole database tables , columns and specific rows without entering the whole value of the searched key was a challenge I had to face.

So how did I manage to get through it?

Well, I had a research on multiple websites that didn't have the exact solution to my problem but they helped me find a solution. I had to set a minimum length of characters that can be written inside the search box so the search function would be able to search effectively. The minimum length was 5 characters. For example the key we are looking for is : "hello". I had to create a query in PHP that was a variable that matched the input value the user would enter. This can be done with `$_GET['query']` and then initialise it to `$query`. Additionally I had to convert the characters inside the `$query` that were readable in HTML but not in PHP with `html special characters` function. Moreover, I added % (percentage signs) in front of the variable but also at the end so the searched key could match related values. For instance in our example "hello" , could match "hellos" , "othello" or even "hello man". Then I had to return the rows that matched the searched key and return them to the website from the database.

But there was one more problem. This problem was the toughest I had to encounter because there was anything like that anywhere and I could not come up with anything on how to implement it but I was thinking of this for a long time, and recently I came up with the solution. The problem was inside the search all page at the right corner of the home page of the publication's website. In the Compile Report I had to combine all the publication types (all the tables in the database) with the columns of authors and years. Needless to say there is in fact no actual query that could fetch all of the above together in rows from the database. On the other hand when you are creative and when you are thinking out of the box you can do anything with the queries in order to fetch specific data . Just because no one else has done it before , it does not mean it is impossible. But what exactly did I do? Well, let's keep in mind that I was searching for it for a long time. So I found a select query that was able to select all tables from the database . But how would I know which tables the user selected? Here come's Javascript. I created a function where I can get the result of every checkbox selected Publication's table which are Books, Journals, Chapters, Conference, Other and Patents. Furthermore, I had to find a away to select author(s) inside those tables. So I divided this in 4 smaller queries which are combined with an "OR" operation in order to select a single author(s), author(s) with common publications and author(s) combined publications and last but not least to select all author(s). Once I had all of those information ,I combined them and then reports are created which then the user could print, and export in different formats.

Chapter 7

Implementation

7.1 Website Overview

Overviewing the website we could see some improvements.

This web content management system is a very useful and convenient tool for a university department to manage its online information. Flexibility and convenience in managing information, user friendly interface and ease of use are the key features that make it a desirable online tool for a department. In addition, with a few minor changes in design, this website can be used by any department at any university. Some valuable extensions which could be done to this website are discussed next.

This Web Content Management System is designed for the faculty at the department of Africana Studies at San Diego State University and so it is required to follow all guidelines for a university website. This website and any information uploaded on it, in any format, must be compliant with web accessibility requirements at all times. So some features like support for uploading documents, pdf files, video files, audio files, etc. would require regular monitoring and maintenance of the website by the concerned web engineers at San Diego State University to ensure compliance with web accessibility requirements at all times. To avoid this extra burden on these engineers, many features have not been supported by this website.

One initially desired feature of this system was support for uploading video and audio files and documents by the faculty members. To support this feature, all video files must have synchronized captions in English and all audio files must have equivalent text transcripts so that they are web accessible. The documents - Microsoft word files, PDFs, etc must be formatted following certain rules to be web accessible. At this point in time, it was not possible to ensure that the uploaded items are compliant with web accessibility requirements at all times and so currently these features are not supported.

As a future enhancement, an uploading interface can be developed that can ensure that documents are in a web accessible format and convert them to such a format if not; before uploading them on the server. Adding support for this feature would be a highly
22

valuable extension to this website. Also, support for features like assignment submission, grade report, etc. would require major revisions, but could be included in this framework.

7.2 High Level View

High Level View of the Website with the Server and Database, among the users/admin of the system and how all are connected.

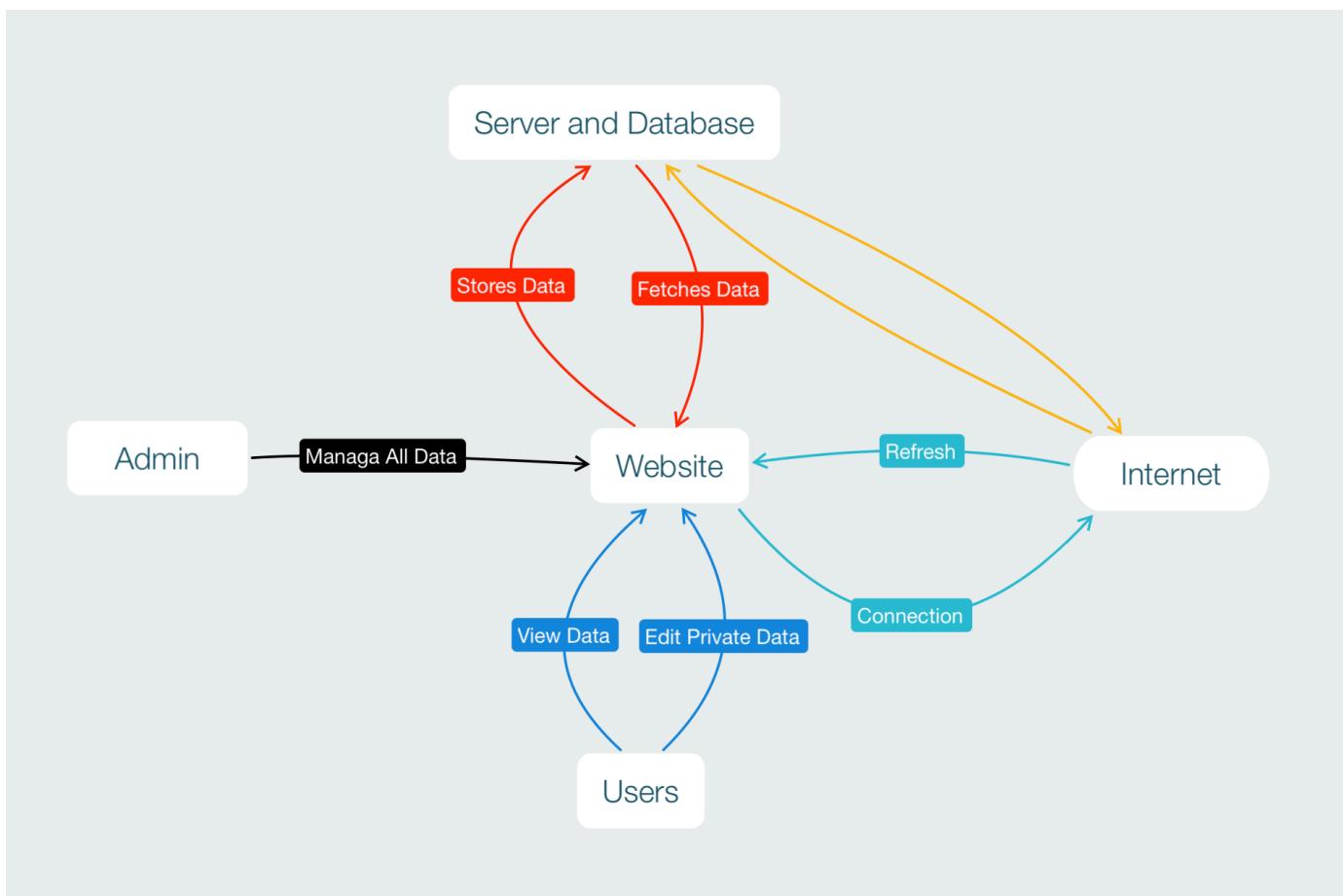


Figure 10 - High Level View

7.3 Compatibility and Accessibility

The website is carefully designed to be user friendly. Its design is very intuitive and informative so that even people with little experience in computers can easily use this system. A user friendly feature of this website is 'title' information for each item in each web page. Moving the mouse pointer over any item such as buttons, text boxes, images, etc will display this title information below the mouse pointer indicating its function or intent. This greatly simplifies interaction with the website as the use of each item on each page becomes known.

Also, the design of the interface through which the faculty will manage their information, is exactly the same as of the interface to the visitors of the website. So while entering information, the faculty will get a close idea of how it will actually look on the website. In

addition, a user manual (will be the Implementation - Chapter 7) with step by step instructions on using this system has also been provided to the department.

7.3.2 WEB ACCESSIBILITY

Web accessibility refers to the inclusive practice of making websites usable by people of all abilities and disabilities. When sites are correctly designed, developed and edited, all users can have equal access to information and functionality. For example, when a site is coded with semantically meaningful HTML, with textual equivalents provided for images and with links named meaningfully, this helps blind users using text-to-speech software and/or text-to-Braille hardware. When click-able links and areas are large, this helps users who cannot control a mouse with precision. When pages are coded so that users can navigate by means of the keyboard alone, or a single switch access device alone, this helps users who cannot use a mouse or even a standard keyboard. When videos are closed captioned or a sign language version is available, deaf and hard of hearing users can understand the video. When flashing effects are avoided or made optional, users prone to seizures caused by these effects are not put at risk. When sites are correctly built and maintained, all of these users can be accommodated while not impacting on the usability of the site for non-disabled users.

This website is carefully designed and implemented so that it meets the web accessibility requirements. All the web pages in the website are verified to meet these requirements by the Web Accessibility Evaluation Tool.

7.3.4 BROWSER COMPATIBILITY

The website is carefully designed and implemented so that it is compatible with different browsers. The website is tested and verified to be working fine on Firefox 5.5 and onwards, Internet Explorer 7 and onwards, Safari 10.1.1 and Google chrome 9.0.552.237.

7.4 Implementation

In this chapter the Implementation shows how the system was built as well as how the user/admin could use this system.

The screenshot shows a navigation bar with 'Publications' on the left and 'Signed in as User' with a 'Log Out' link on the right. Below this is a dark header bar with 'Publications' on the left and a 'Search' button on the right. A green navigation bar below contains links for 'Books', 'Journals', 'Books Chapters', 'Conferences', 'Patents', and 'Other'. The main content area is titled 'Web-based Database System Publications'.

This is a website where the publications are going to be published. Each user will have different privileges and can see his own publications or search for his publications. Also for administrator privileges there will be an option to view and search and publish all publications. Additionally after the user enters each of those options, there will be a new page where there is first going to be a text box asking for the name of the publication . If the publication already exists in the system the rest of the fields are going to be added automatically. Note: During the import settings there must be an option where the user have the option to replace completely overwrite to the latest version of the publication or import a new one. All of these publication are going to be added dynamically and you should be able to preview them at tables. To the system there is going to be only one admin (secretary) but there are going to be several users who can be added dynamically after signing up.

Users can:

- 1.submit publications
- 2.be connected with the system and there should be a menu where publications are going to be added
- 3.preview all the publications
- 4.search an author and find publications
- 5.author should be able to get a report with the publications with categories(books table, conferences etc)
- 6.be able to view relation with other author
- 7.search all page

This is the Home screen of the proposed system which consists of seven buttons. In order to go to the desired screen, the users just have to log in first.

7.4.1 Log In

Before entering into the system (home page), the user must login, for this purpose the log in page is created.

Publications

Login

Email

Your Email

Password

Your Password

Login

New User? [Sign Up Here](#)

This log in form is made for security purpose i.e. only authenticated users have access into the system, i.e. either administrator or the user. So let's zoom in order to see clearly how the login page, looks like. The user has to enter his email and password in order to log in the system.

Login

Email

Your Email

Password

Your Password

Login

New User? [Sign Up Here](#)

Don't have **an account?**
Please proceed below.

Sign Up

Name

Email

Password

Confirm Password

[Sign Up](#)

Already Registered? [Login Here](#)

What happens if you are a new user?

Well, you can click the “Sign Up here” which is shown above, and it will direct you to the sign up page shown below. This will create an account and your information will be sent to the database. Let’s have a closer look at the sign up page.

7.4.2 Sign Up Page

At the sign up page you have to fill your name, email and you have to enter your password twice in order to create an account.

Sign Up

Name

Email

Password

Confirm Password

[Sign Up](#)

Already Registered? [Login Here](#)

Well Done!

It was so simple.

Now you have an account at the system.

Once you follow the above instructions , let's proceed in order to see what else you can do with the system.

NOW You should be able to view the menu below.

7.4.3 Menu

The screenshot shows a user interface for a 'Web-based Database System Publications' application. At the top, there is a navigation bar with 'Publications' on the left and 'Signed in as User' with a 'Log Out' link on the right. Below the navigation bar is a dark header bar with 'Publications' on the left and a 'Search' button on the right. The header bar is divided into six green-colored buttons labeled 'Books', 'Journals', 'Books Chapters', 'Conferences', 'Patents', and 'Other'. The main content area displays the title 'Web-based Database System Publications'.

So,

The **menu** is divided in 6 **buttons** which are:

1.BOOKS

2.JOURNALS

3.BOOK CHAPTERS

4.CONFERENCES

5.PATENTS

6.OTHER

Try to click for example the **JOURNALS** button.

What do you see?

7.4.4 Journals Menu



User has 5 buttons again.

So let's view what each button does.

Add - Add Journal Form to the system

Delete - Delete Journal Title to the system

Search - Update - Give's the ability to search for a Journal Title and update/edit any information of that record

Back To Menu - Takes you BACK to the menu of the home page.

7.4.5 Journal Add

Add = > Add button let's you create a Journal's Form and insert the follow information to the system.

Journals Form

*For the required fields

Author(s) *

Paper Title *

Journal name *

Volume *

Number *

Pages Length *

Date *

Submit

Back

So, for example the user can enter:

Author: Polykarpou and P. Ioannou

Paper Title: A Robust Adaptive Nonlinear Control Design

Journal Name: Automatica

Volume: 2

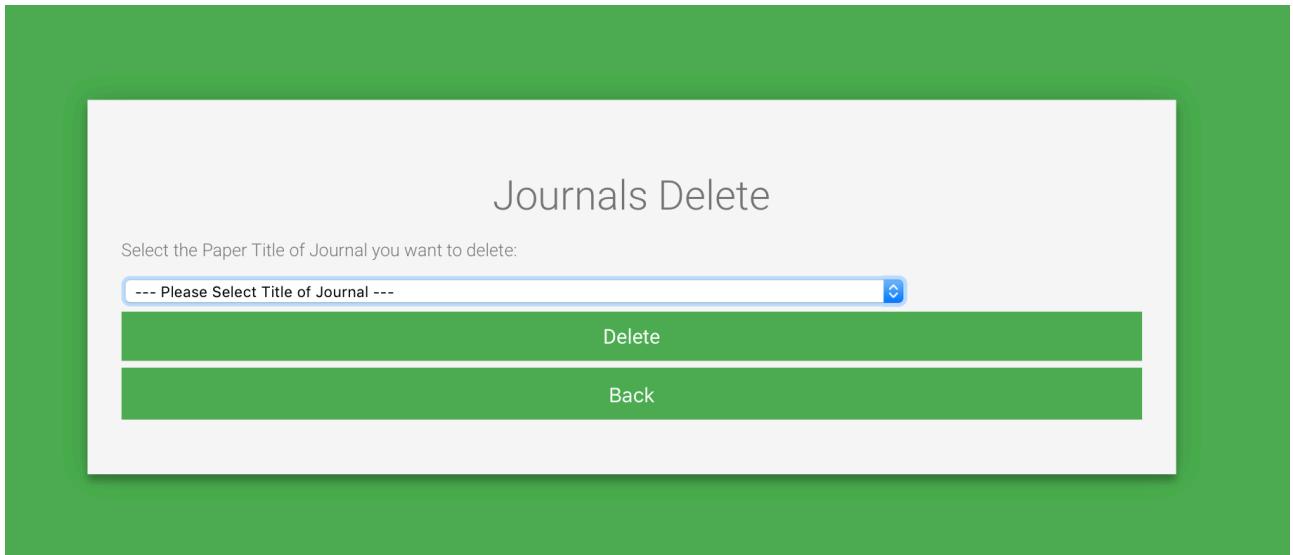
Number: 3

Pages: 423-427

Date: 1996-03-01

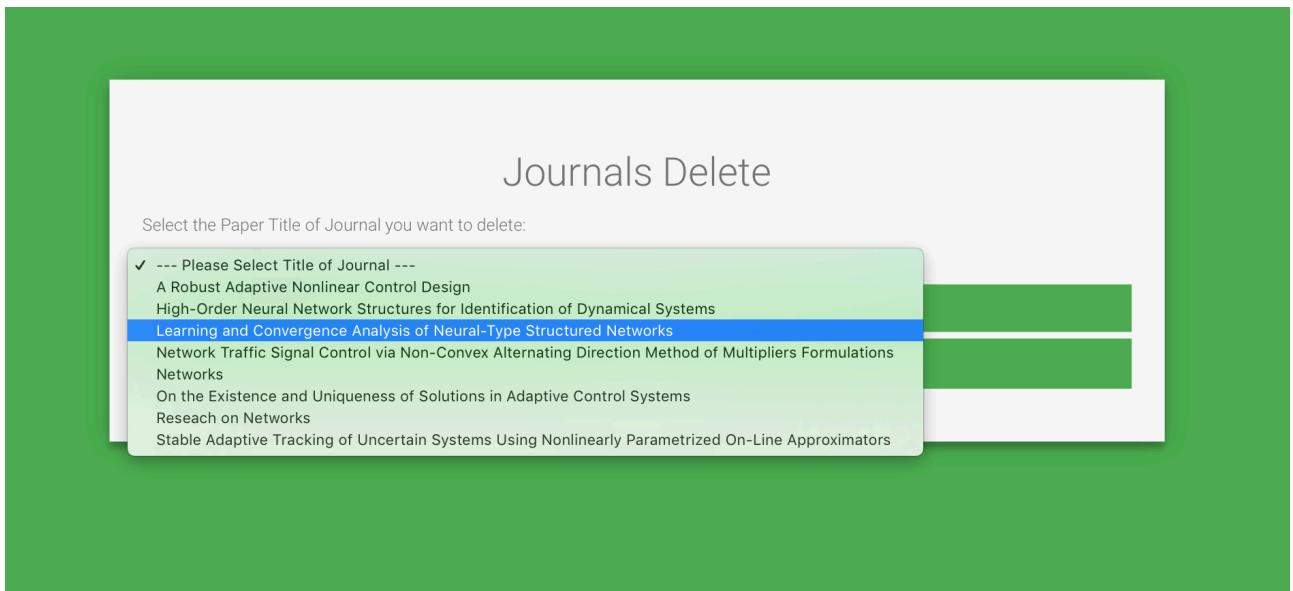
And then click **SUBMIT** button in order to add the above information to the system.

7.4.6 Journals Delete



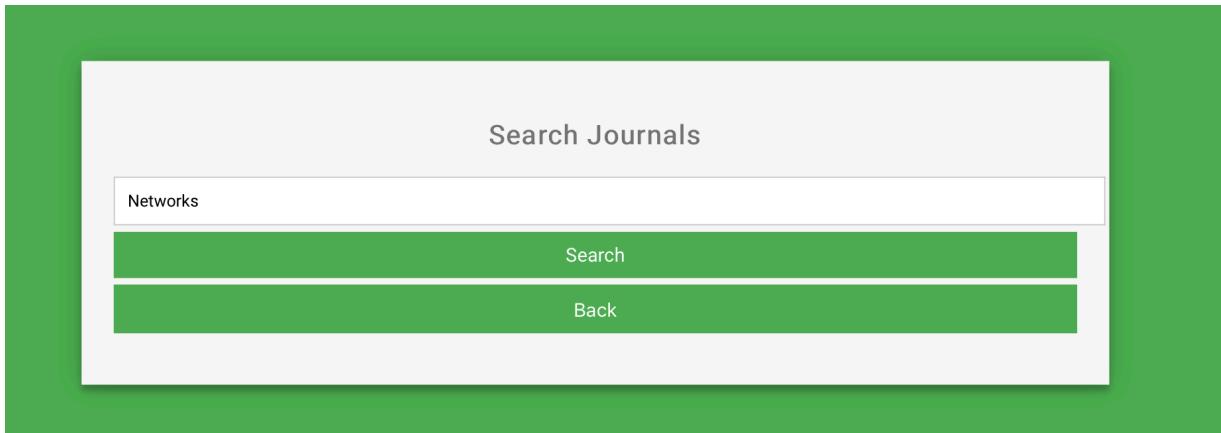
Journals Delete is shown above. So the user has to select the Title of Journal in order to delete the entire record inside the database.

So let's say the user selects '**Learning and Convergence Analysis of Neural-Type Structured Networks'**



The record will delete almost instantly the record (entire row) inside the database and the webpage will be refreshed.

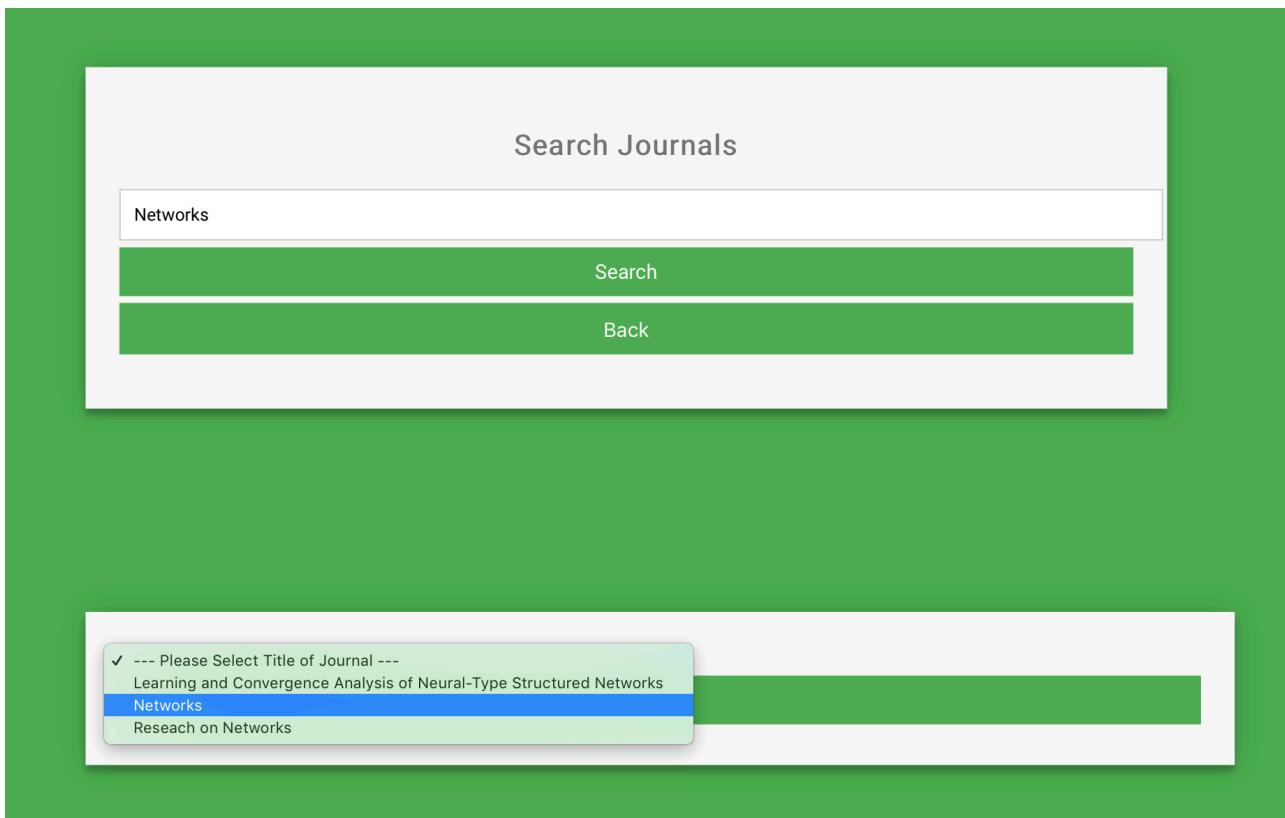
7.4.7 Journals Search - Update



In this webpage the user has the ability to search the entire table of Journals **by author** or **by journal title**. Then this will give the user the ability to edit/update a specific record inside the database.

But there are hundred of thousand of records inside the database, so the user doesn't remember the entire title of the Journal . This is no problem for the system because it can **dynamically search** any related values the user enters inside the search box , and it will display all the similar Journals Title.

Here's the preview:



The user searches for "**Networks**"

As we can see the system fetches all the similar values that are inside the database.

So the user has to select which one to edit/update in order to proceed to the webpage where the user will update or edit the specific record. So below we can see the entire record and edit the values we want EXCEPT the title because it is the only way to relate the record to the database.

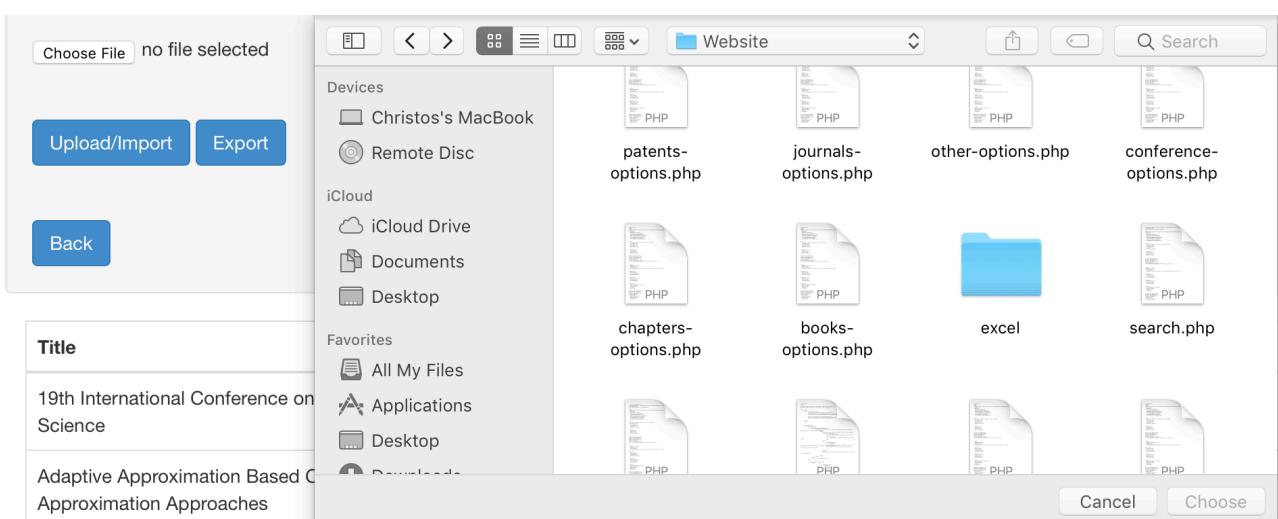
Journals Form Updated

Title:	<input type="text" value="Networks"/>
Author:	<input type="text" value="Polycarpou"/>
Journal:	<input type="text" value="Networking"/>
Volume:	<input type="text" value="19"/>
Number:	<input type="text" value="11"/>
Pages:	<input type="text" value="19-150"/>
Date:	<input type="text" value="2009-10-10"/>
Edit/Update	

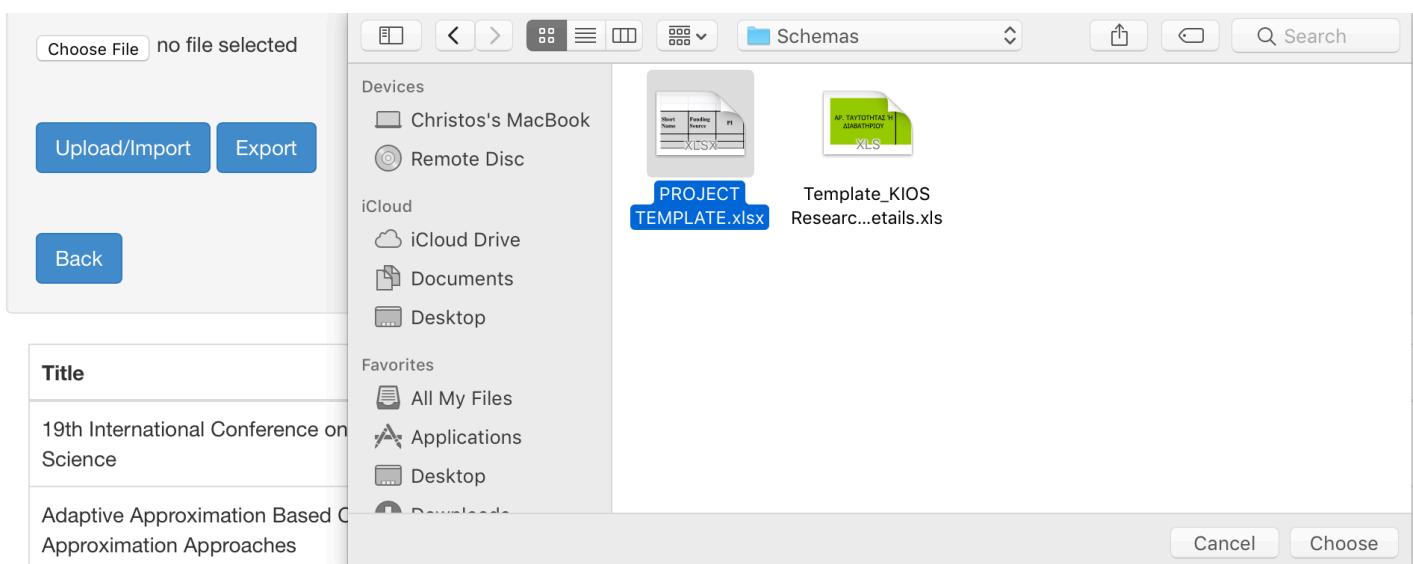
After the user edits the record and clicks the Edit button the result will be as shown below.

Now the record is updated inside the database. And the next step for the user is to Create A new Journal or Go Back to the central home page Menu or search again with the Update Journal.

7.4.8 Import and Export



The system has the ability to **IMPORT AND EXPORT** csv or excel files. The user has to choose a file from the computer that is the format of .xls .xlsx or .csv.



And then after the user clicks choose and then Upload/Import the data should be displayed like below.

Preview after the user has imported or exported to the system.

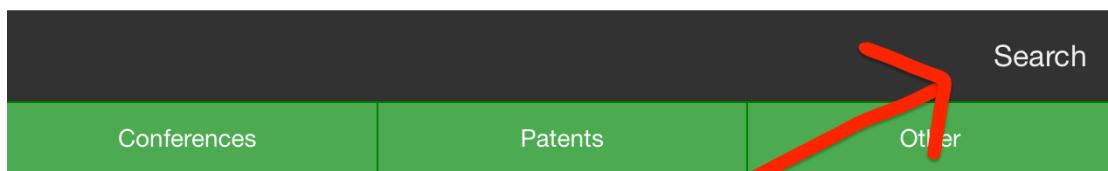
Import CSV/Excel file for Journals

no file selected

Paper Title	Author	Journal	Volume	Number	Pages	Journal Date
A Robust Adaptive Nonlinear Control Design	Polycarpou and P. Ioannou	Automatica	32	3	423-427	1996-03-01
High-Order Neural Network Structures for Identification of Dynamical Systems	Kosmatopoulos, M. Polycarpou, M. Christodoulou, and P. Ioannou	IEEE Transactions on Neural Networks	16	4	422-431	1995-01-03
Learning and Convergence Analysis of Neural-Type Structured Networks	Polycarpou and P. Ioannou	IEEE Transactions on Neural Networks	4	10	39-50	2017-01-03
Network Traffic Signal Control via Non-Convex Alternating Direction Method of Multipliers Formulations	Timotheou, C.G. Panayiotou and M. Polycarpou	Transportation Research Record: Traffic Flow Theory and Characteristics	2490	1	11-20	2015-01-01
Networks	Polycarpou	Networking	19	11	19-150	2009-10-10
On the Existence and Uniqueness of Solutions in Adaptive Control Systems	Polycarpou and P. Ioannou	IEEE Transactions on Automatic Control	38	3	474-479	1993-01-03
Research on Networks	Polycarpou	IEEE Networks	19	2	9-154	2009-01-10
Stable Adaptive Tracking of Uncertain Systems Using Nonlinearly Parametrized On-Line Approximators	Polycarpou and M. Mears	International Journal of Control	70	3	363-384	1998-05-01

7.4.9 Search (All in database) - Compile Report

The user has the ability to search the entire database **tables, authors and date** with the button. It is located in the upper right corner as shown below.



System Publications

Let's break **Compile Report** into 3 SEARCH SECTIONS

Compile Report

Report Options:

1. Select any combination of publication type:

- Books
- Journals
- Book Chapters
- Conferences
- Patents
- Other
- Select All

First of all the user has to choose the publication type (which refers to the table inside the database).

2. Select any combination of author(s): (Please fill ONLY 1 OPTION of the following 4 options)

2.1 Search for single author:

eg. Polycarpou

OR

2.2 Search for author(s) common publication(s):

eg. Liu H. Zhang M. Polycarpou

OR

2.3 Search for author(s) combined publication(s):

eg. C. Panayiotou and G. Ellinas

OR

2.4 Search for all author(s) in the database: Select All

Secondly you choose the authors.

There are 4 options for the authors because this gives the opportunity to search for a single author(s) publications or common publications or author(s) combined publications or even all the author(s) in the database with just a "Select All" checkbox.

*Note: The user has to choose ONLY one option of the above, otherwise the system will show an alert which won't allow the user to continue.

3. Select any combination of year(s) range:

Date From:

Date eg.Y-M-D-> 2001-09-25

Date To:

Date eg.Y-M-D-> 2012-10-26

Search

Back

Last but not least, the user has to choose the year(s) range.

Date MUST BE IN **YEAR - MONTH -DAY FORMAT**

Otherwise, a warning error will inform the user the entered value is **INCORRECT**.

Back to Main Page

Export in CSV

Export in PDF

Print

FINALLY the user can click search and can preview the results.

Furthermore, the user has the ability to generate REPORTS and export them in different formats such as CSV or PDF. Also the user has the ability to PRINT them with the printer.

7.5 Code

DBCONNECT.PHP

```
<?php
    //connect to mysql database
    $con = mysqli_connect("localhost", "root", "9667", "kios") or die("Error " .
    mysqli_error($con));
?
?>
```

REGISTER.PHP

```
<?php
session_start();

if(isset($_SESSION['usr_id'])) {
    header("Location: index.php");
}

include_once 'dbconnect.php';

//set validation error flag as false
$error = false;

//check if form is submitted
if (isset($_POST['signup'])) {
    $name = mysqli_real_escape_string($con, $_POST['name']);
    $email = mysqli_real_escape_string($con, $_POST['email']);
    $password = mysqli_real_escape_string($con, $_POST['password']);
    $cpassword = mysqli_real_escape_string($con, $_POST['cpassword']);

    //name can contain only alpha characters and space
    if (!preg_match("/^([a-zA-Z ])+$/,$name)) {
        $error = true;
        $name_error = "Name must contain only alphabets and space";
    }
    if(!filter_var($email,FILTER_VALIDATE_EMAIL)) {
        $error = true;
        $email_error = "Please Enter Valid Email ID";
    }
    if(strlen($password) < 6) {
        $error = true;
        $password_error = "Password must be minimum of 6 characters";
    }
    if($password != $cpassword) {
        $error = true;
    }
}
```

```

$cpassword_error = "Password and Confirm Password doesn't match";
}
if (!$error) {
    if(mysqli_query($con, "INSERT INTO users(name,email,password) VALUES('"
. $name . "','" . $email . "','" . md5($password) . "')")) {
        $successmsg = "Successfully Registered! <a href='login.php'>Click
here to Login</a>";
    } else {
        $errmsg = "Error in registering...Please try again later!";
    }
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>User Registration Script</title>
    <meta content="width=device-width, initial-scale=1.0" name="viewport" >
    <link rel="stylesheet" href="css/bootstrap.min.css" type="text/css" />
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
    <div class="container-fluid">
        <!-- add header -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target="#navbar1">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="index.php">KIOS</a>
        </div>
        <!-- menu items -->
        <div class="collapse navbar-collapse" id="navbar1">
            <ul class="nav navbar-nav navbar-right">
                <li><a href="login.php">Login</a></li>
                <li class="active"><a href="register.php">Sign Up</a></li>
            </ul>
        </div>
    </div>
</nav>

```

```

<div class="container">
    <div class="row">
        <div class="col-md-4 col-md-offset-4 well">
            <form role="form" action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post" name="signupform">
                <fieldset>
                    <legend>Sign Up</legend>

                    <div class="form-group">
                        <label for="name">Name</label>
                        <input type="text" name="name" placeholder="Enter Full Name" required value="<?php if($error) echo $name; ?>" class="form-control" />
                        <span class="text-danger"><?php if (isset($name_error)) echo $name_error; ?></span>
                    </div>

                    <div class="form-group">
                        <label for="email">Email</label>
                        <input type="text" name="email" placeholder="Email" required value="<?php if($error) echo $email; ?>" class="form-control" />
                        <span class="text-danger"><?php if (isset($email_error)) echo $email_error; ?></span>
                    </div>

                    <div class="form-group">
                        <label for="password">Password</label>
                        <input type="password" name="password" placeholder="Password" required class="form-control" />
                        <span class="text-danger"><?php if (isset($password_error)) echo $password_error; ?></span>
                    </div>

                    <div class="form-group">
                        <label for="cpassword">Confirm Password</label>
                        <input type="password" name="cpassword" placeholder="Confirm Password" required class="form-control" />
                        <span class="text-danger"><?php if (isset($cpassword_error)) echo $cpassword_error; ?></span>
                    </div>

                    <div class="form-group">

```

```

        <input type="submit" name="signup"
value="Sign Up" class="btn btn-primary" />
            </div>
        </fieldset>
    </form>
    <span class="text-success"><?php if (isset($successmsg)) { echo
$successmsg; } ?></span>
    <span class="text-danger"><?php if (isset($errmsg)) { echo
$errmsg; } ?></span>
            </div>
        </div>
        <div class="row">
            <div class="col-md-4 col-md-offset-4 text-center">
                Already Registered? <a href="login.php">Login Here</a>
            </div>
        </div>
    </div>
<script src="js/jquery-1.10.2.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>

```

LOGIN.PHP

```

<?php
session_start();

if(isset($_SESSION['usr_id'])!=" ") {
    header("Location: index.php");
}

include_once 'dbconnect.php';

//check if form is submitted
if (isset($_POST['login'])) {

    $email = mysqli_real_escape_string($con, $_POST['email']);
    $password = mysqli_real_escape_string($con, $_POST['password']);
    $result = mysqli_query($con, "SELECT * FROM users WHERE email = '" . $email. "'"
and password = '" . md5($password) . "'");

    if ($row = mysqli_fetch_array($result)) {

```

```

$_SESSION['usr_id'] = $row['id'];
$_SESSION['usr_name'] = $row['name'];
header("Location: index.php");
} else {
    $errmsg = "Incorrect Email or Password!!!";
}
?>

<!DOCTYPE html>
<html>
<head>
    <title>PHP Login Script</title>
    <meta content="width=device-width, initial-scale=1.0" name="viewport" >
    <link rel="stylesheet" href="css/bootstrap.min.css" type="text/css" />
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
    <div class="container-fluid">
        <!-- add header -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target="#navbar1">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="index.php">Publications</a>
        </div>
        <!-- menu items -->
        <div class="collapse navbar-collapse" id="navbar1">
            <ul class="nav navbar-nav navbar-right">
                <li class="active"><a href="login.php">Login</a></li>
                <li><a href="register.php">Sign Up</a></li>
            </ul>
        </div>
    </div>
</nav>

<div class="container">
    <div class="row">
        <div class="col-md-4 col-md-offset-4 well">

```

```

<form role="form" action="<?php echo $_SERVER['PHP_SELF']; ?>" method="post" name="loginform">
    <fieldset>
        <legend>Login</legend>

        <div class="form-group">
            <label for="name">Email</label>
            <input type="text" name="email" placeholder="Your Email" required class="form-control" />
        </div>

        <div class="form-group">
            <label for="name">Password</label>
            <input type="password" name="password" placeholder="Your Password" required class="form-control" />
        </div>

        <div class="form-group">
            <input type="submit" name="login" value="Login" class="btn btn-primary" />
        </div>
    </fieldset>
</form>
<span class="text-danger"><?php if (isset($errmsg)) { echo $errmsg; } ?></span>
</div>
</div>
<div class="row">
    <div class="col-md-4 col-md-offset-4 text-center">
        New User? <a href="register.php">Sign Up Here</a>
    </div>
</div>
</div>

```

```

<script src="js/jquery-1.10.2.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>

```

LOGOUT.PHP

```

<?php
session_start();
//comment on log out

if(isset($_SESSION['usr_id'])) {

```

```

session_destroy();
unset($_SESSION['usr_id']);
unset($_SESSION['usr_name']);
header("Location: index.php");
} else {
    header("Location: index.php");
}
?>

```

MENU.PHP

```

<!DOCTYPE html>
<html>
<head>
<style>
body {margin:0;}
ul.topnav {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #333;
}

/*ul.topnav li {float: left;}*/
ul.topnav li {float: right ;}

ul.topnav li a {
    display: inline-block;
    color: #f2f2f2;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
    transition: 0.3s;
    font-size: 17px;
}

ul.topnav li a:hover {background-color: #555;}

ul.topnav li.icon {display: none;}

@media screen and (max-width:680px) {
    ul.topnav li:not(:first-child) {display: none;}
    ul.topnav li.icon {

```

```

float: right;
display: inline-block;
}
}

@media screen and (max-width:680px) {
ul.topnav.responsive {position: relative;}
ul.topnav.responsive li.icon {
position: absolute;
right: 0;
top: 0;
}
ul.topnav.responsive li {
float: none;
display: inline;
}
ul.topnav.responsive li a {
display: block;
text-align: left;
}
}
</style>
</head>
<body>

<ul class="topnav" id="myTopnav">
<div style="float:left;">
<li><a class="active" href="#home">Publications</a></li>
</div>
<!-- <li><a href="#news">News</a></li>-->
<!--<li><a href="#contact">Contact</a></li>-->
<li><a href="#about" onclick="location.href='/all-search.php'">Search</a></li>
<li class="icon">
<a href="javascript:void(0);" style="font-size:15px;" onclick="myFunction()">≡</a>
</li>
</ul>

<style>
.btn-group button {
background-color: #4CAF50; /* Green background */
border: 1px solid green; /* Green border */
color: white; /* White text */

```

```

padding: 10px 24px; /* Some padding */
cursor: pointer; /* Pointer/hand icon */
float: left; /* Float the buttons side by side */
}

/* Clear floats (clearfix hack) */
.btn-group:after {
content: " ";
clear: both;
display: table;
}

.btn-group button:not(:last-child) {
border-right: none; /* Prevent double borders */
}

/* Add a background color on hover */
.btn-group button:hover {
background-color: #3e8e41;
}

</style>
<body>

<!--<h1>Justified Button Group</h1>-->

<!--<p>Four buttons in a group:</p>-->

<div class="btn-group" style="width:100%">
<button style="width:16.5%" onclick="location.href='/books-options.php'">Books</button>
<button style="width:16.5%" onclick="location.href='/journals-options.php'">Journals</button>
<button style="width:17%" onclick="location.href='/chapters-options.php'">Books Chapters</button>
<button style="width:17%" onclick="location.href='/conference-options.php'">Conferences</button>
<button style="width:16.5%" onclick="location.href='/patents-options.php'">Patents</button>
<button style="width:16.5%" onclick="location.href='/other-options.php'">Other</button>
</div>

<script>
function myFunction() {

```

```

var x = document.getElementById("myTopnav");
if (x.className === "topnav") {
    x.className += " responsive";
} else {
    x.className = "topnav";
}
}

</script>

</body>
</html>

```

INDEX.PHP

```

<?php
session_start();
include_once 'dbconnect.php';
//new code
//include 'upload.php';
//if(isset($_SESSION['usr_id'])) {
//include 'services.php';
//}

//end of new code

?>
<!DOCTYPE html>
<html>
<head>
    <title>Home </title>
    <meta content="width=device-width, initial-scale=1.0" name="viewport" >
        <link rel="stylesheet" href="css/bootstrap.min.css" type="text/css" />
</head>
<body>

<nav class="navbar navbar-default" role="navigation">
    <div class="container-fluid">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target="#navbar1">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
        </div>
    <div class="collapse navbar-collapse" id="navbar1">
        <ul class="nav navbar-nav">
            <li class="active"><a href="#">Home</a></li>
            <li><a href="#">About</a></li>
            <li><a href="#">Services</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </div>
</div>
</nav>
<div class="container" style="margin-top: 20px;">
    <div class="row">
        <div class="col-md-6" style="background-color: #f2f2f2; padding: 10px; border-radius: 5px; margin-bottom: 10px;">
            <h3>About Us</h3>
            <p>We are a team of professionals dedicated to providing high-quality services. Our mission is to make your life easier by offering reliable and efficient solutions. We believe in transparency, accountability, and customer satisfaction. We are always looking for ways to improve and stay ahead in our field. Thank you for choosing us!</p>
        </div>
        <div class="col-md-6" style="background-color: #f2f2f2; padding: 10px; border-radius: 5px; margin-bottom: 10px;">
            <h3>Our Services</h3>
            <ul style="list-style-type: none; padding-left: 0;">
                <li><a href="#">Service A</a></li>
                <li><a href="#">Service B</a></li>
                <li><a href="#">Service C</a></li>
                <li><a href="#">Service D</a></li>
            </ul>
        </div>
    </div>
</div>

```

```
<a class="navbar-brand" href="index.php">Publications</a>
</div>
<div class="collapse navbar-collapse" id="navbar1">
    <ul class="nav navbar-nav navbar-right">
        <?php if (isset($_SESSION['usr_id'])) { ?>
            <li><p class="navbar-text">Signed in as <?php echo
$_SESSION['usr_name']; ?></p></li>
            <li><a href="logout.php">Log Out</a></li>
            <?php } else { ?>
            <li><a href="login.php">Login</a></li>
            <li><a href="register.php">Sign Up</a></li>
            <?php } ?>
        </ul>
    </div>
</div>
</body>

<body> <?php if (isset($_SESSION['usr_id'])) {
include 'menu.php';
} ?>
</body>

<body>

<!-- New Code starts from here -->
<!--<div>-->

    <div class="item-page" itemscope="" itemtype="http://schema.org/
Article">
<meta itemprop="inLanguage" content="en-GB">

        <div class="page-header">
            <h2 style="text-align: center;" itemprop="name"> Web-based Database System
Publications </h2>
            <!-- <p style="text-align: center;"><span style="font-
size: 10pt;"><em></em></span></p>-->

        </div>
        <div itemprop="articleBody">
<p style="text-align: center;"></p>
<p style="text-align: left;">This is a website where the publications are going to be
published. Each user will have different privileges and can see his own publications or
search for his
```

publications. Also for administrator privileges there will be an option to view and search and publish all publications. Additionally after the user enters each of those options, there will be

a new page where there is first going to be a text box asking for the name of the publication . If the publication already exists in the system the rest of the fields are going to be added automatically.

Note: During the import settings there must be an option where the user have the option to replace completely overwrite to the latest version of the publication or import a new one. All of these publication are going to be added

dynamically and you should be able to preview them at tables.

To the system there is going to be only one admin (secretary) but there are going to be several users who can be added dynamically after signing up.</p>

 Users can:</br>

 1.submit publications

 2.be connected with the system and there should be a menu where publications are going to be added

</br>

 3.preview all the publications

 4.search an author and find publications

</br>

 5.author should be able to get a report with the publications with categories(books table, conferences etc)

 6.be able to view relation with other author

</br>

 7.search all page

</div>

<div id="fav-copyright" class="clearfix">

<div id="fav-showcopyright" class="span12">

</div>

</div>

</body>

<!-- Codes endes here-->

```
<script src="js/jquery-1.10.2.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

BOOKS-ADD.PHP

```
<!DOCTYPE html>
<html>
<style>
@import url(https://fonts.googleapis.com/css?family=Roboto:400,300,600,400italic);
margin: 0;
padding: 0;
box-sizing: border-box;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
-webkit-font-smoothing: antialiased;
-moz-font-smoothing: antialiased;
-o-font-smoothing: antialiased;
font-smoothing: antialiased;
text-rendering: optimizeLegibility;
}
```

```
body {
font-family: "Roboto", Helvetica, Arial, sans-serif;
font-weight: 100;
font-size: 12px;
line-height: 30px;
color: #777;
background: #4CAF50;
}
```

```
.container {
max-width: 800px;
width: 100%;
margin: 0 auto;
position: relative;
}
```

```
#contact input[type="text"],
#contact input[type="email"],
#contact input[type="tel"],
```

```
#contact input[type="url"],  
#contact textarea,  
#contact button[type="submit"] {  
    font: 400 12px/16px "Roboto", Helvetica, Arial, sans-serif;  
}  
  
#contact {  
    background: #F9F9F9;  
    padding: 25px;  
    margin: 150px 0;  
    box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);  
}  
  
#contact h3 {  
    display: block;  
    font-size: 30px;  
    font-weight: 300;  
    margin-bottom: 10px;  
}  
  
#contact h4 {  
    margin: 5px 0 15px;  
    display: block;  
    font-size: 13px;  
    font-weight: 400;  
}  
  
fieldset {  
    border: medium none !important;  
    margin: 0 0 10px;  
    min-width: 100%;  
    padding: 0;  
    width: 100%;  
}  
  
#contact input[type="text"],  
#contact input[type="email"],  
#contact input[type="tel"],  
#contact input[type="url"],  
#contact textarea {  
    width: 100%;  
    border: 1px solid #ccc;  
    background: #FFF;  
    margin: 0 0 5px;
```

```
padding: 10px;  
}  
  
#contact input[type="text"]:hover,  
#contact input[type="email"]:hover,  
#contact input[type="tel"]:hover,  
#contact input[type="url"]:hover,  
#contact textarea:hover {  
    -webkit-transition: border-color 0.3s ease-in-out;  
    -moz-transition: border-color 0.3s ease-in-out;  
    transition: border-color 0.3s ease-in-out;  
    border: 1px solid #aaa;  
}  
  
#contact textarea {  
    height: 100px;  
    max-width: 100%;  
    resize: none;  
}  
  
#contact button[type="submit"] {  
    cursor: pointer;  
    width: 100%;  
    border: none;  
    background: #4CAF50;  
    color: #FFF;  
    margin: 0 0 5px;  
    padding: 10px;  
    font-size: 15px;  
}  
  
#contact button[type="submit"]:hover {  
    background: #43A047;  
    -webkit-transition: background 0.3s ease-in-out;  
    -moz-transition: background 0.3s ease-in-out;  
    transition: background-color 0.3s ease-in-out;  
}  
  
#contact button[type="submit"]:active {  
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);  
}  
  
/*NEW CODE HERE*/
```

```
#contact button[type="button"] {  
    cursor: pointer;  
    width: 100%;  
    border: none;  
    background: #4CAF50;  
    color: #FFF;  
    margin: 0 0 5px;  
    padding: 10px;  
    font-size: 15px;  
}  
  
#contact button[type="button"]:hover {  
    background: #43A047;  
    -webkit-transition: background 0.3s ease-in-out;  
    -moz-transition: background 0.3s ease-in-out;  
    transition: background-color 0.3s ease-in-out;  
}  
  
#contact button[type="button"]:active {  
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);  
}  
/*END OF NEW CODE*/  
  
.copyright {  
    text-align: center;  
}  
  
#contact input:focus,  
#contact textarea:focus {  
    outline: 0;  
    border: 1px solid #aaa;  
}  
  
::-webkit-input-placeholder {  
    color: #888;  
}  
  
:-moz-placeholder {  
    color: #888;  
}  
  
::-moz-placeholder {  
    color: #888;  
}
```

```

:-ms-input-placeholder {
    color: #888;
}
</style>
<body>

<?php
// define variables and set to empty values

// title author publisher places pages years

$titleErr = $authorErr = $publisherErr = $placesErr = $pagesErr = $yearsErr = "";
$title = $author = $publisher = $places = $pages = $years = "";

function validateDate($date, $format = 'Y-m-d')
{
    $d = DateTime::createFromFormat($format, $date);
    return $d && $d->format($format) == $date;
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["Author"])) {
        $authorErr = "Author Name is required";
    } else {
        $author = test_input($_POST["Author"]);
        // check if name only contains letters and whitespace
        if (!preg_match('/^([a-zA-Zñéáíóú]+)$', $author)) {
            $authorErr = "Only letters and white space allowed";
            echo '<script language="javascript">';
            echo 'alert("Incorrect Author Format")';
            echo '</script>';
        }
    }
}

if (empty($_POST["Title"])) {
    $titleErr = "Title is required";
} else {
    $title = test_input($_POST["Title"]);
    // check if name only contains letters and whitespace
    if (!preg_match('/^([a-zA-Zñéáíóú]+)$', $title)) {
        // if (!preg_match('/^([a-zA-Zñéáíóú]+)$', $paper)) {

```

```

$titleErr = "Only letters and white space allowed";
echo '<script language="javascript">';
echo 'alert("Incorrect Title")';
echo '</script>';
}

}

if (empty($_POST["Publisher"])) {
    $publisherErr = "Publisher is required";
} else {
    $publisher = test_input($_POST["Publisher"]);
// check if name only contains letters and whitespace
if (!preg_match('/^a-zA-Z\s\.\w+$/i', $publisher)) {
    $publisherErr = "Only letters and white space allowed";
    echo '<script language="javascript">';
    echo 'alert("Incorrect Publisher")';
    echo '</script>';
}
}

if (empty($_POST["Places"])) {
    $placesErr = "Places is required";
} else {
    $places = test_input($_POST["Places"]);
if (!preg_match('/^a-zA-Z\s\.\w+$/i', $places)) {
    $placesErr = "Only letters and white space allowed";
    echo '<script language="javascript">';
    echo 'alert("Incorrect Places")';
    echo '</script>';
}
}

if (empty($_POST["Pages"])) {
    $pagesErr = "Pages number is required";
} else {
    $pages = test_input($_POST["Pages"]);
// check if name only contains letters and whitespace
//if (!preg_match("/^a-zA-Z\w+$/i", $Pages)) {
// if (!preg_match('/^0-9\w+$/i', $Pages)) {

if (!preg_match('/^0-9\w+$/i', $pages)) {

```

```

$pagesErr = "Only a range of number and dash (-) is allowed";
echo '<script language="javascript">';
echo 'alert("Incorrect Pages Format")';
echo '</script>';
}

}

if (empty($_POST["Years"])) {
    $yearsErr = "Year is required";
} else {
    $years = (int) $years;

    $years = test_input($_POST["Years"]);

    if ($years<1900 || $years>2100) {
        //invalid
        $yearsErr = "Year must be less than 2100 and greater than 1900";
        echo '<script language="javascript">';
        echo 'alert("Incorrect Year.")';
        echo '</script>';
    }
}
}

function test_input($data)
{
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>

<div class="container">
<form id="contact" method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">

<h3 align="center">Books Form</h3>
<h4>*For the required fields</h4>

```

```

<fieldset>
    <span> <?php echo 'Author(s)';?> </span>
        <span class="error">* <?php echo $AuthorErr;?></span>
        <input placeholder="Enter Author(s)" type="text" name="Author" tabindex="1" <?
    php if (!empty($_POST['Author'])) {
        echo "value=\"" . $_POST["Author"] . "\"";
    } ?> required autofocus>

</fieldset>

<fieldset>
    <span> <?php echo 'Title';?> </span>
        <span class="error">* <?php echo $titleErr;?></span>
        <input placeholder="Enter title ..." type="text" name="Title" tabindex="2" <?php if
    (!empty($_POST['Title'])) {
        echo "value=\"" . $_POST["Title"] . "\"";
    } ?> required autofocus>
<!-- <span class="error">* <?php echo $titleErr;?></span>-->
</fieldset>

<fieldset>
    <span> <?php echo 'Publisher';?> </span>
        <span class="error">* <?php echo $publisherErr;?></span>
            <input placeholder="Enter publisher..." type="text" name="Publisher"
    tabindex="3" <?php if (!empty($_POST['Publisher'])) {
        echo "value=\"" . $_POST["Publisher"] . "\"";
    } ?> required autofocus>
</fieldset>

<fieldset>
    <span> <?php echo 'Places';?> </span>
        <span class="error">* <?php echo $placesErr;?></span>
        <input placeholder="Enter places ..." type="text" name="Places" tabindex="4" <?
    php if (!empty($_POST['Places'])) {
        echo "value=\"" . $_POST["Places"] . "\"";
    } ?>required autofocus>

</fieldset>

<fieldset>
    <span> <?php echo 'Pages Length';?> </span>
        <span class="error">* <?php echo $PagesErr;?></span>
        <input placeholder="Pages Length (eg. pp.10-55)" type="text" name="Pages"
    tabindex="6" <?php if (!empty($_POST['Pages'])) {

```

```
echo "value=\"\" . $_POST["Pages"] . "\";  
} ?>required autofocus>  
  
</fieldset>  
  
  
<fieldset>  
    <span> <?php echo 'Years';?> </span>  
    <span class="error">* <?php echo $yearsErr;?></span>  
    <input placeholder="Year YYYY eg. 2001" type="text" name ="Years"  
tabindex="7" <?php if (!empty($_POST['Years'])) {  
    echo "value=\"\" . $_POST["Years"] . "\";  
} ?>required autofocus>  
  
</fieldset>  
  
  
<fieldset>  
    <button name="submit" type="submit" value="Submit" id="contact-submit" data-  
submit="...Sending">Submit</button>  
    <!-- <button name="back" value="Back" onclick='menu.php'>Back</button>-->  
    <!-- working one <a class=".back_button" type="submit" id="contact-submit"  
href="javascript:history.back(1)">Back</a>-->  
  
</fieldset>  
  
  
<fieldset>  
    <!--<button name="submit" type="submit" value="Submit" id="contact-submit"  
onclick="location.href='/options.php'">Back</button>-->  
    <button id="contact-submit" type="button" onclick="window.location.href='/books-  
options.php'"> Back </button>  
  
  
</fieldset>  
  
</form>  
  
  
</div>  
  
  
  
  
<?php
```

```

if (isset($_POST['submit'])) {
    /*echo "<h2>Your Input:</h2>";

echo $Date;
echo "<br>";*/

$servername = "localhost";
$username = "root";
$password = "9667";
$dbname = "kios";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

//New Code starts here///

$title = test_input($_POST["Title"]);

$sql2 = "SELECT * FROM books where (title = '$title')";

/*
if ($result=mysqli_query($con,$sql))
{
// Return the number of rows in result set
$rowcount=mysqli_num_rows($result);
printf("Result set has %d rows.\n",$rowcount);
// Free result set
mysqli_free_result($result);
}

-----
$dupesql = "SELECT * FROM table where (name = '$name' AND description = '$description' AND manufacturer = '$manufacturer' AND city ='$city' AND price = '$price' AND enddate = '$end_date')";

$duperaw = mysql_query($dupesql);

if (mysql_num_rows($duberaw) > 0) {

```

```

//your code ...
}

*/
$result2 = mysqli_query($conn, $sql2);
$rowcount= mysqli_num_rows($result2);

if ($rowcount > 0) {
    /*
echo '<script language="javascript">';
echo "Error: " . $sql2 . "<br>" . $conn->error;
echo 'alert("Error:Duplicate Value!");';
echo '</script>'*/



echo '<script language="javascript">';
    echo 'alert("Title already Exists")';
    echo '</script>';
} else {
    $sql = "INSERT INTO books (title,author,publisher,places,pages,years) VALUES
('{$title}', '{$author}', '{$publisher}', '{$places}', '{$pages}', '{$years}')";

    if ($conn->query($sql) === true) {
        echo '<script language="javascript">';
        echo 'alert("New record created successfully")';
        echo '</script>';
    } else {
        echo '<script language="javascript">';
        echo "Error: " . $sql . "<br>" . $conn->error;
        echo 'alert("Error:Incorrect values added. Please try again!")';
        echo '</script>';
    }
}

$conn->close();
}

?>

</body>
</html>

```

BOOKS-DELETE.PHP

```
<!DOCTYPE html>
<html>
<style>
@import url(https://fonts.googleapis.com/css?family=Roboto:400,300,600,400italic);
margin: 0;
padding: 0;
box-sizing: border-box;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
-webkit-font-smoothing: antialiased;
-moz-font-smoothing: antialiased;
-o-font-smoothing: antialiased;
font-smoothing: antialiased;
text-rendering: optimizeLegibility;
}

body {
font-family: "Roboto", Helvetica, Arial, sans-serif;
font-weight: 100;
font-size: 12px;
line-height: 30px;
color: #777;
background: #4CAF50;
}

.container {
max-width: 800px;
width: 100%;
margin: 0 auto;
position: relative;
}

#contact input[type="text"],
#contact input[type="email"],
#contact input[type="tel"],
#contact input[type="url"],
#contact textarea,
#contact button[type="submit"] {
font: 400 12px/16px "Roboto", Helvetica, Arial, sans-serif;
}

#contact {
```

```
background: #F9F9F9;
padding: 25px;
margin: 150px 0;
box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}

#contact h3 {
display: block;
font-size: 30px;
font-weight: 300;
margin-bottom: 10px;
}

#contact h4 {
margin: 5px 0 15px;
display: block;
font-size: 13px;
font-weight: 400;
}

fieldset {
border: medium none !important;
margin: 0 0 10px;
min-width: 100%;
padding: 0;
width: 100%;
}

#contact input[type="text"],
#contact input[type="email"],
#contact input[type="tel"],
#contact input[type="url"],
#contact textarea {
width: 100%;
border: 1px solid #ccc;
background: #FFF;
margin: 0 0 5px;
padding: 10px;
}

#contact input[type="text"]:hover,
#contact input[type="email"]:hover,
#contact input[type="tel"]:hover,
#contact input[type="url"]:hover,
#contact textarea:hover {
```

```
-webkit-transition: border-color 0.3s ease-in-out;  
-moz-transition: border-color 0.3s ease-in-out;  
transition: border-color 0.3s ease-in-out;  
border: 1px solid #aaa;  
}
```

```
#contact textarea {  
height: 100px;  
max-width: 100%;  
resize: none;  
}
```

```
#contact button[type="submit"] {  
cursor: pointer;  
width: 100%;  
border: none;  
background: #4CAF50;  
color: #FFF;  
margin: 0 0 5px;  
padding: 10px;  
font-size: 15px;  
}
```

```
#contact button[type="submit"]:hover {  
background: #43A047;  
-webkit-transition: background 0.3s ease-in-out;  
-moz-transition: background 0.3s ease-in-out;  
transition: background-color 0.3s ease-in-out;  
}
```

```
#contact button[type="submit"]:active {  
box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);  
}
```

*/*NEW CODE HERE*/*

```
#contact button[type="button"] {  
cursor: pointer;  
width: 100%;  
border: none;  
background: #4CAF50;  
color: #FFF;  
margin: 0 0 5px;  
padding: 10px;
```

```
font-size: 15px;  
}  
  
#contact button[type="button"]:hover {  
    background: #43A047;  
    -webkit-transition: background 0.3s ease-in-out;  
    -moz-transition: background 0.3s ease-in-out;  
    transition: background-color 0.3s ease-in-out;  
}  
  
#contact button[type="button"]:active {  
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);  
}  
/*END OF NEW CODE*/  
  
.copyright {  
    text-align: center;  
}  
  
#contact input:focus,  
#contact textarea:focus {  
    outline: 0;  
    border: 1px solid #aaa;  
}  
  
:::-webkit-input-placeholder {  
    color: #888;  
}  
  
:-moz-placeholder {  
    color: #888;  
}  
  
::-moz-placeholder {  
    color: #888;  
}  
  
:-ms-input-placeholder {  
    color: #888;  
}  
}  
</style>  
<head>  
<title>Delete Data</title>  
</head>  
<body>
```

```
<div class="container">

<form id="contact" name="form_update" method="post" action="books-delete.php">

<h3 align="center">Books Delete</h3>

<fieldset>

<?php
$con=mysqli_connect("localhost","root","9667","kios");
//===== check connection
if(mysqli_errno($con))
{
    echo "Can't Connect to mySQL:".mysqli_connect_error();
}
else
{
    echo "Select the Title of Book you want to delete: <br>";
}

//=====
//This creates the drop down box
echo "<select name='title'>";
echo '<option value="">.'--- Please Select Title of Book ---'.'</option>';
//$query=mysqli_query($con,"SELECT id,FirstName FROM persons");
$query = mysqli_query($con,"SELECT title FROM books");
$query_display = mysqli_query($con,"SELECT * FROM books");
while($row=mysqli_fetch_array($query))
{
    echo "<option value='".$row['title']."'>".$row['title']
    .'</option>';

}

echo '</select>';
?>
<!--<button id="contact-submit" type="submit" name="submit" value="Delete"/>-->
```

```

<button name="submit" type="submit" value="Delete" id="contact-submit">Delete</button>
<button id="contact-submit" type="button" onclick="window.location.href='/books-options.php'"> Back </button>
</fieldset>
</form>
</div>
<!--
<br/><br/>

<button id="contact-submit" type="submit" onclick="location.href='/index.php'"> Menu </button>

-->

<fieldset id="contact" class="container" >

<?php
$con=mysqli_connect("localhost","root","9667","kios");
if(mysqli_errno($con))
{
    echo "Can't Connect to mySQL:".mysqli_connect_error();
}
// $name = mysqli_real_escape_string($con,$_POST['select']);
// $fetch = mysqli_query($con,"SELECT * FROM persons WHERE FirstName='".$name."'");
// $row_display=mysqli_fetch_assoc($fetch);
if(isset($_POST['title']))
{
    $title = $_POST['title'];
    // $name = mysqli_real_escape_string($con,$_POST['select']);
    // $fetch = "SELECT * FROM persons WHERE FirstName = '".$name."'";
    // $fetch="SELECT 'Firstname' FROM persons WHERE Firstname = '".$name."'";
    //FETCH WORKING ONE
    // $fetch="SELECT paper_title,author,journal,volume,numb,pages,journal_date FROM journals WHERE paper_title = '".$name."'";
    //DELETE FROM `conference` WHERE `conference`.`title` = '\'DDCC\'?

    $delete = "DELETE FROM books WHERE title = '".$title."'";
}

```

```

$result = mysqli_query($con,$delete);
if(!$result)
{
echo "Error:".(mysqli_error($con));
}
//display the table
else
{
if(isset($_POST['submit']))
{
echo "<meta http-equiv='refresh' content='0'>";
}
echo "Data $title Deleted successfully!";
}

/*
// In Database Table Journals => paper_title author journal volume numb pages
journal_date
echo '<table border="1">'. '<tr>'. '<td align="center">'. 'From Database'. '</td>'. '</tr>';
//echo '<tr>'. '<td>'. '<table border="1">'. '<tr>'. '<td>'. 'First Name'. '</td>'. '<td>'. 'Last
Name'. '</td>'. '<td>'. 'Gender' . '</td>'. '<td>'. 'Subject'. '</td>'. '<td>'. 'Hobbies' . '</
td>'. '</tr>';
echo '<tr>'. '<td>'. '<table border="1">'. '<tr>'. '<td>'. 'Paper Title'. '</
td>'. '<td>'. 'Author'. '</td>'. '<td>'. 'Journal' . '</td>'. '<td>'. 'Volume'. '</td>'. '<td>'.
'Number' . '</td>'. '<td>'. 'Pages'. '</td>'. '<td>'. 'Journal Date' . '</td>'. '</tr>';

//while($data = mysqli_fetch_row($fetch))
while($data=mysqli_fetch_row($result))
{
echo ("<tr><td>$data[0]</td><td>$data[1]</td><td>$data[2]</td><td>$data[3]</
td><td>$data[4]</td><td>$data[5]</td><td>$data[6]</td></tr>");
}
echo '</table>'. '</td>'. '</tr>'. '</table>';*/
}

?>

```

```

<?php

$con=mysql_connect("localhost", "root", "9667") or die("Could not connect");
mysql_select_db("kios",$con) or die("could not connect database");
?>

<!--
<div>
<table class="table table-bordered">
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Journal</th>
<th>Volume</th>
<th>Number</th>
<th>Pages</th>
<th>Journal Date</th>

</tr>
</thead>-->
<?php

echo '<table border="1">'. '<tr>'. '<td align="center">'. 'From Database'. '</td>'. '</tr>';
echo '<tr>'. '<td>'. '<table border="1">'. '<tr>'. '<td>'. 'Title'. '</td>'. '<td>'. 'Author'. '</td>'. '<td>'. 'Publisher' . '</td>'. '<td>'. 'Places'. ' '. '</td>'. '<td>'. 'Pages'. '</td>'. '<td>'. 'Years' . '</td>'. '</tr>';

$SQLSELECT = "SELECT * FROM books";
$result_set = mysql_query($SQLSELECT, $con);
while($row = mysql_fetch_array($result_set))
{
?>

<tr>
<td><?php echo $row['title']; ?></td>
<td><?php echo $row['author']; ?></td>
<td><?php echo $row['publisher']; ?></td>
<td><?php echo $row['places']; ?></td>
<td><?php echo $row['pages']; ?></td>
<td><?php echo $row['years']; ?></td>

```

```

// In Database Table Journals => paper_title author journal volume numb pages
journal_date
    echo '<table border="1">.<tr>.<td align="center">'. 'From Database'. '</
td>.</tr>';
        //echo '<tr>.<td>.<table border="1">.<tr>.<td>.'First Name'. '</
td>.<td>.'Last Name'. '</td>.<td>.'Gender' .</td>.<td>.'Subject'. '</td>.<td>'.
'Hobbies' .</td>.</tr>';
            echo '<tr>.<td>.<table border="1">.<tr>.<td>.'Paper Title'. '</
td>.<td>.'Author'. '</td>.<td>.'Journal' .</td>.<td>.'Volume'. '</td>.<td>'.
'Number' .</td>.<td>.'Pages'.</td>.<td>.'Journal Date' .</td>.</tr>';
//while($data = mysqli_fetch_row($fetch))
while($data=mysqli_fetch_row($result))
{
    echo ("<tr><td>$data[0]</td><td>$data[1]</td><td>$data[2]</
td><td>$data[3]</td><td>$data[4]</td><td>$data[5]</td><td>$data[6]</td></tr>");
}
echo '</table>.</td>.</tr>.</table>'-->
</tr>
<?php
}
echo '</table>.</td>.</tr>.</table>';
?>
<!--</table>

</div>-->
</fieldset>

</body>
</html>

```

BOOKS-DISPLAY.PHP

```
<!DOCTYPE html>
<html>
<style>
@import url(https://fonts.googleapis.com/css?family=Roboto:400,300,600,400italic);
margin: 0;
padding: 0;
box-sizing: border-box;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
-webkit-font-smoothing: antialiased;
-moz-font-smoothing: antialiased;
-o-font-smoothing: antialiased;
font-smoothing: antialiased;
text-rendering: optimizeLegibility;
}

body {
font-family: "Roboto", Helvetica, Arial, sans-serif;
font-weight: 100;
font-size: 12px;
line-height: 30px;
color: #777;
background: #4CAF50;
}

.container {
max-width: 800px;
width: 100%;
margin: 0 auto;
position: relative;
}

#contact input[type="text"],
#contact input[type="email"],
#contact input[type="tel"],
#contact input[type="url"],
#contact textarea,
#contact button[type="submit"] {
font: 400 12px/16px "Roboto", Helvetica, Arial, sans-serif;
}

#contact {
background: #F9F9F9;
```

```
padding: 25px;
margin: 150px 0;
box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}

#contact h3 {
display: block;
font-size: 30px;
font-weight: 300;
margin-bottom: 10px;
}

#contact h4 {
margin: 5px 0 15px;
display: block;
font-size: 13px;
font-weight: 400;
}

fieldset {
border: medium none !important;
margin: 0 0 10px;
min-width: 100%;
padding: 0;
width: 100%;
}

#contact input[type="text"],
#contact input[type="email"],
#contact input[type="tel"],
#contact input[type="url"],
#contact textarea {
width: 100%;
border: 1px solid #ccc;
background: #FFF;
margin: 0 0 5px;
padding: 10px;
}

#contact input[type="text"]:hover,
#contact input[type="email"]:hover,
#contact input[type="tel"]:hover,
#contact input[type="url"]:hover,
#contact textarea:hover {
-webkit-transition: border-color 0.3s ease-in-out;
```

```
-moz-transition: border-color 0.3s ease-in-out;  
transition: border-color 0.3s ease-in-out;  
border: 1px solid #aaa;  
}  
  
  
#contact textarea {  
height: 100px;  
max-width: 100%;  
resize: none;  
}  
  
  
#contact button[type="submit"] {  
cursor: pointer;  
width: 100%;  
border: none;  
background: #4CAF50;  
color: #FFF;  
margin: 0 0 5px;  
padding: 10px;  
font-size: 15px;  
}  
  
  
#contact button[type="submit"]:hover {  
background: #43A047;  
-webkit-transition: background 0.3s ease-in-out;  
-moz-transition: background 0.3s ease-in-out;  
transition: background-color 0.3s ease-in-out;  
}  
  
  
#contact button[type="submit"]:active {  
box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);  
}  
  
  
.copyright {  
text-align: center;  
}  
  
  
#contact input:focus,  
#contact textarea:focus {  
outline: 0;  
border: 1px solid #aaa;  
}  
  
  
::-webkit-input-placeholder {  
color: #888;
```

```

}

:-moz-placeholder {
  color: #888;
}

::-moz-placeholder {
  color: #888;
}

:-ms-input-placeholder {
  color: #888;
}

</style>

```

<?php

```

$connection = mysql_connect('localhost','root','9667') or die ("Couldn't connect to
server.");
$db = mysql_select_db('kios', $connection) or die ("Couldn't select database.");
// In Database Table Journals => paper_title author journal volume numb pages
journal_date -->
$key=$_POST['keyfield'];
$title=$_POST['Title'];
$author=$_POST['Author'];
$publisher=$_POST['Publisher'];
$places=$_POST['Places'];
$pages=$_POST['Pages'];
$years=$_POST['Years'];

// $data = "UPDATE `table_name` SET name='$Name', age='$Age', hobby='Hobby'
WHERE ID=". $$ . $key . "'";
$data = "UPDATE `books` SET author='$author', publisher ='$publisher' , places
 ='$places', pages = '$pages', years = '$years' WHERE title=". $$ . $title . "'";
$query = mysql_query($data) or die("Couldn't execute query. ". mysql_error());

```

?>

```

<head>
  <title>Updated</title>
</head>

```

```
<body>

<div class="container">

    <form id="contact" action="books-display.php">
        <h3 align="center">Updated Books Form</h3>
        <h4>Displaying Changed Record Sent</h4>

        <fieldset>
            <!-- display the changed record from database -->
            Title: <?php echo $title?><br>
            Author: <?php echo $author ?> <br>
            Publisher: <?php echo $publisher ?> <br>
            Places: <?php echo $places ?> <br>
            Pages: <?php echo $pages ?> <br>
            Places: <?php echo $places ?> <br>
            Years: <?php echo $years ?> <br>
        </fieldset>
    </form>
    <fieldset id= "contact">
        <button style = "button" type="submit" onclick="location.href='/books-add.php'">
        Create New Journal </button>

        <button style = "button" type="submit" onclick="location.href='/index.php'"> Menu
    </button>

        <button type="submit" onclick="location.href='/books-search.php'"> Update Books</button>
    </fieldset>

</div>
    <!-- WOrking one <a href="journals.php"> Create New Journal</a>
        <a href="index.php"> Menu </a>
        <a href="fetch.php"> Update Journal </a> -->

</body>

</html>
```

BOOKS-EXPORT.PHP

```

<?php

// output headers so that the file is downloaded rather than displayed
header('Content-Type: text/csv; charset=utf-8');
header('Content-Disposition: attachment; filename=data.csv');

// create a file pointer connected to the output stream
$output = fopen('php://output', 'w');

// output the column headings
fputcsv($output, array('Title', 'Author', 'Publisher', 'Places', 'Pages', 'Years'));

// fetch the data
mysql_connect('localhost', 'root', '9667');
mysql_select_db('kios');
/*Paper Title Author Journal Volume Number Pages Journal Date*/
$rows = mysql_query('SELECT title,author,publisher,places,pages,years FROM books');

// loop over the rows, outputting them
while ($row = mysql_fetch_assoc($rows)) fputcsv($output, $row);

//New Code Starts Here
//Exported data
/*
$con=mysqli_connect("localhost","root","9667","kios");
if(mysqli_errno($con))
{
    echo "Can't Connect to mySQL:".mysqli_connect_error();
}
//$/name = mysqli_real_escape_string($con,$_POST['select']);
// $fetch = mysqli_query($con,"SELECT * FROM persons WHERE FirstName='".$
$name."'");
// $row_display=mysqli_fetch_assoc($fetch);
//$/name = mysqli_real_escape_string($con,$_POST['select']);
//fetch = "SELECT * FROM persons WHERE FirstName = '".$name."'";
//fetch="SELECT 'Firstname' FROM persons WHERE Firstname = '".$name."'";
$fetch="SELECT title,author,conference,pages,conference_date FROM conference";
$result = mysqli_query($con,$fetch);
if(!$result)

```

```

{
echo "Error:".mysqli_error($con));
}
//display the table

// In Database Table Journals => paper_title author journal volume numb pages
journal_date
echo '<table border="1">'. '<tr>'. '<td align="center">'. 'From Database'. '</td>'. '</tr>';
//echo '<tr>'. '<td>'. '<table border="1">'. '<tr>'. '<td>'. 'First Name'. '</td>'. '<td>'. 'Last
Name'. '</td>'. '<td>'. 'Gender' . '</td>'. '<td>'. 'Subject'. '</td>'. '<td>'. 'Hobbies' . '</
td>'. '</tr>';
echo '<tr>'. '<td>'. '<table border="1">'. '<tr>'. '<td>'. 'Paper Title'. '</
td>'. '<td>'. 'Author'. '</td>'. '<td>'. 'Journal' . '</td>'. '<td>'. 'Volume'. '</td>'. '<td>'.
'Number' . '</td>'. '<td>'. 'Pages'. '</td>'. '<td>'. 'Journal Date' . '</td>'. '</tr>';

//while($data = mysqli_fetch_row($fetch))
while($data=mysqli_fetch_row($result))
{
echo ("<tr><td>$data[0]</td><td>$data[1]</td><td>$data[2]</td><td>$data[3]</
td><td>$data[4]</td><td>$data[5]</td><td>$data[6]</td></tr>");
}
echo '</table>'. '</td>'. '</tr>'. '</table>';
*/
?>

```

BOOKS-IMPORT-NEXT.PHP

<?php

```

$con=mysql_connect("localhost","root","9667") or die("Could not connect");
mysql_select_db("kios",$con) or die("could not connect database");
if(isset($_POST["Import"])){

```

```
echo $filename=$_FILES["file"]["tmp_name"];
```

```

if($_FILES["file"]["size"] > 0)
{

```

```

$file = fopen($filename, "r");
while (($emapData = fgetcsv($file, 10000, ",")) !== FALSE)
{
    // $emapData[4] = date('Y-m-d',
strtotime($emapData[4]));
    // echo $emapData[5];

    $sql = "INSERT into books (`title`, `author`, `publisher`, `places`, `pages`,
`years`)

values('$emapData[0]', '$emapData[1]', '$emapData[2]', '$emapData[3]', '$emapData[4]', '$e
mapData[5]')";
    //we are using mysql_query function. it returns a resource on true else False
on error
    $result = mysql_query( $sql, $con );

    if( ! $result )
    {
        echo "<script type=\"text/javascript\">
            alert(\"Invalid File:Please Upload CSV File.
\"");
        window.location = \"books-import.php\"
        </script>";

    }

}

fclose($file);
//throws a message if data successfully imported to mysql database from
excel file
echo "<script type=\"text/javascript\">
    alert(\"CSV File has been successfully Imported.
\"");
    window.location = \"books-import.php\"
    </script>";

//close of connection
mysql_close($con);

```

```

        }
    }
?>
BOOKS-IMPORT.PHP

<!DOCTYPE html>
<?php

$con=mysql_connect("localhost","root","9667") or die("Could not connect");
mysql_select_db("kios",$con) or die("could not connect database");

?>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>Import Excel To Mysql Database Using PHP </title>
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta name="description" content="Import Excel File To MySql Database
Using php">

        <link rel="stylesheet" href="css/bootstrap.min.css">
        <link rel="stylesheet" href="css/bootstrap-responsive.min.css">
        <link rel="stylesheet" href="css/bootstrap-custom.css">

    </head>
    <body>

        <div id="wrap">
            <div class="container">
                <div class="row">
                    <div class="span3 hidden-phone"></div>
                    <div class="span6" id="form-login">
                        <form class="form-horizontal well" action="books-import-
next.php" method="post" name="upload_excel" enctype="multipart/form-data">
                            <fieldset>
                                <legend align= "center">Import CSV/Excel file
for Books</legend>
                                <div class="control-group">
                                    <div class="control-label">
                                        <!--<label align= "center">CSV/
Excel File:</label>-->
                                    </div>

```

```

        <div class="controls">
            <input type="file" name="file">
<id="file" class="input-large">
        </div>
    </div>

    <div class="control-group">
        <div class="controls">
            <br><br>
            <button type="submit" id="submit"
name="Import" class="btn btn-primary button-loading" data-loading-
text="Loading...">Upload/Import</button>
            <a type="submit" class="btn btn-primary
button-loading" href="books-export.php"> Export</a>
        </div>
    </div>
    <br><br>
    <a type="submit" class="btn btn-primary
button-loading" href="books-options.php">Back</a>

        </fieldset>
    </form>
</div>
<div class="span3 hidden-phone"></div>
</div>

<table class="table table-bordered">
    <thead>
        <tr>
            <th>Title</th>
            <th>Author</th>
            <th>Publisher</th>
            <th>Places</th>
            <th>Pages</th>
            <th>Years</th>
        </tr>
    </thead>
<?php
    $SQLSELECT = "SELECT * FROM books";
    $result_set = mysql_query($SQLSELECT, $con);

```

```
while($row = mysql_fetch_array($result_set))
{
?>

<tr>
    <td><?php echo $row['title']; ?></td>
    <td><?php echo $row['author']; ?></td>
    <td><?php echo $row['publisher']; ?></td>
    <td><?php echo $row['places']; ?></td>
    <td><?php echo $row['pages']; ?></td>
    <td><?php echo $row['years']; ?></td>

</tr>
<?php
}
?>
</table>
</div>

</div>

</body>
</html>
```

BOOKS-OPTIONS.PHP

```
<!DOCTYPE html>
<html>
<style>
@import url(https://fonts.googleapis.com/css?family=Roboto:400,300,600,400italic);
margin: 0;
padding: 0;
box-sizing: border-box;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
-webkit-font-smoothing: antialiased;
-moz-font-smoothing: antialiased;
-o-font-smoothing: antialiased;
font-smoothing: antialiased;
text-rendering: optimizeLegibility;
}

body {
```

```
font-family: "Roboto", Helvetica, Arial, sans-serif;
```

```
font-weight: 100;
font-size: 12px;
line-height: 30px;
color: #777;
background: #4CAF50;
}

.container {
  max-width: 800px;
  width: 100%;
  margin: 0 auto;
  position: relative;
}

#contact input[type="text"],
#contact input[type="email"],
#contact input[type="tel"],
#contact input[type="url"],
#contact textarea,
#contact button[type="submit"] {
  font: 400 12px/16px "Roboto", Helvetica, Arial, sans-serif;
}

#contact {
  background: #F9F9F9;
  padding: 25px;
  margin: 150px 0;
  box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}

#contact h3 {
  display: block;
  font-size: 30px;
  font-weight: 300;
  margin-bottom: 10px;
}

#contact h4 {
  margin: 5px 0 15px;
  display: block;
  font-size: 13px;
  font-weight: 400;
}
```

```
fieldset {  
    border: medium none !important;  
    margin: 0 0 10px;  
    min-width: 100%;  
    padding: 0;  
    width: 100%;  
}  
  
#contact input[type="text"],  
#contact input[type="email"],  
#contact input[type="tel"],  
#contact input[type="url"],  
#contact textarea {  
    width: 100%;  
    border: 1px solid #ccc;  
    background: #FFF;  
    margin: 0 0 5px;  
    padding: 10px;  
}  
  
#contact input[type="text"]:hover,  
#contact input[type="email"]:hover,  
#contact input[type="tel"]:hover,  
#contact input[type="url"]:hover,  
#contact textarea:hover {  
    -webkit-transition: border-color 0.3s ease-in-out;  
    -moz-transition: border-color 0.3s ease-in-out;  
    transition: border-color 0.3s ease-in-out;  
    border: 1px solid #aaa;  
}  
  
#contact textarea {  
    height: 100px;  
    max-width: 100%;  
    resize: none;  
}  
  
#contact button[type="submit"] {  
    cursor: pointer;  
    width: 100%;  
    border: none;  
    background: #4CAF50;  
    color: #FFF;  
    margin: 0 0 5px;  
    padding: 10px;  
}
```

```
font-size: 15px;  
}  
  
#contact button[type="submit"]:hover {  
    background: #43A047;  
    -webkit-transition: background 0.3s ease-in-out;  
    -moz-transition: background 0.3s ease-in-out;  
    transition: background-color 0.3s ease-in-out;  
}  
  
#contact button[type="submit"]:active {  
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);  
}  
  
.copyright {  
    text-align: center;  
}  
  
#contact input:focus,  
#contact textarea:focus {  
    outline: 0;  
    border: 1px solid #aaa;  
}  
  
::-webkit-input-placeholder {  
    color: #888;  
}  
  
:-moz-placeholder {  
    color: #888;  
}  
  
::-moz-placeholder {  
    color: #888;  
}  
  
:-ms-input-placeholder {  
    color: #888;  
}  
}
```

```
<head>
    <title>Books Menu</title>
</head>

<body>
    <div class="container">

        <fieldset id= "contact">
            <h3 align="center">Books Menu</h3>

            <button type="submit" onclick="location.href='books-add.php'"> Add
</button>

            <button type="submit" onclick="location.href='/books-delete.php'">
Delete </button>

            <button type="submit" onclick="location.href='/books-search.php'">
Search - Update</button>

            <button type="submit" onclick="location.href='/books-import.php'" >
Preview - Import - Export </button>

            <button type="submit" onclick="location.href='/index.php'" >Back To
Menu </button>

        </fieldset>

    </div>

</body>

</html>
```

BOOKS-SEARCH.PHP

```
<!-- New Code Added Here -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<style>
@import url(https://fonts.googleapis.com/css?family=Roboto:400,300,600,400italic);
margin: 0;
padding: 0;
box-sizing: border-box;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
-webkit-font-smoothing: antialiased;
-moz-font-smoothing: antialiased;
-o-font-smoothing: antialiased;
font-smoothing: antialiased;
text-rendering: optimizeLegibility;
}

body {
font-family: "Roboto", Helvetica, Arial, sans-serif;
font-weight: 100;
font-size: 14px;
line-height: 30px;
color: #777;
background: #4CAF50;
}

.container {
max-width: 800px;
width: 100%;
margin: 0 auto;
position: relative;
}

#contact input[type="text"],
#contact input[type="email"],
#contact input[type="tel"],
#contact input[type="url"],
#contact textarea,
#contact button[type="submit"] {
```

```
font: 400 12px/16px "Roboto", Helvetica, Arial, sans-serif;  
}
```

```
#contact {  
background: #F9F9F9;  
padding: 25px;  
margin: 150px 0;  
box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);  
}
```

```
#contact h3 {  
display: block;  
font-size: 30px;  
font-weight: 300;  
margin-bottom: 10px;  
}
```

```
#contact h4 {  
margin: 5px 0 15px;  
display: block;  
font-size: 13px;  
font-weight: 400;  
}
```

```
fieldset {  
border: medium none !important;  
margin: 0 0 10px;  
min-width: 100%;  
padding: 0;  
width: 100%;  
}
```

```
#contact input[type="text"],  
#contact input[type="email"],  
#contact input[type="tel"],  
#contact input[type="url"],  
#contact textarea {  
width: 100%;  
border: 1px solid #ccc;  
background: #FFF;  
margin: 0 0 5px;  
padding: 10px;  
}
```

```
#contact input[type="text"]:hover,  
#contact input[type="email"]:hover,  
#contact input[type="tel"]:hover,  
#contact input[type="url"]:hover,  
#contact textarea:hover {  
    -webkit-transition: border-color 0.3s ease-in-out;  
    -moz-transition: border-color 0.3s ease-in-out;  
    transition: border-color 0.3s ease-in-out;  
    border: 1px solid #aaa;  
}  
  
#contact textarea {  
    height: 100px;  
    max-width: 100%;  
    resize: none;  
}  
  
#contact button[type="submit"] {  
    cursor: pointer;  
    width: 100%;  
    border: none;  
    background: #4CAF50;  
    color: #FFF;  
    margin: 0 0 5px;  
    padding: 10px;  
    font-size: 15px;  
}  
  
#contact button[type="submit"]:hover {  
    background: #43A047;  
    -webkit-transition: background 0.3s ease-in-out;  
    -moz-transition: background 0.3s ease-in-out;  
    transition: background-color 0.3s ease-in-out;  
}  
  
#contact button[type="submit"]:active {  
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);  
}  
  
/*NEW CODE HERE*/  
  
#contact button[type="button"] {  
    cursor: pointer;  
    width: 100%;  
    border: none;
```

```
background: #4CAF50;
color: #FFF;
margin: 0 0 5px;
padding: 10px;
font-size: 15px;
}

#contact button[type="button"]:hover {
background: #43A047;
-webkit-transition: background 0.3s ease-in-out;
-moz-transition: background 0.3s ease-in-out;
transition: background-color 0.3s ease-in-out;
}

#contact button[type="button"]:active {
box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);
}
/*END OF NEW CODE*/

.copyright {
text-align: center;
}

#contact input:focus,
#contact textarea:focus {
outline: 0;
border: 1px solid #aaa;
}

::webkit-input-placeholder {
color: #888;
}

:-moz-placeholder {
color: #888;
}

::-moz-placeholder {
color: #888;
}

:-ms-input-placeholder {
color: #888;
}
```

```

</style>
<head>
    <title>Search</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <!--<link rel="stylesheet" type="text/css" href="style.css"/>-->
</head>
<body>
    <div class="container">
        <form id = "contact" action="books-search.php" method="GET">
            <h2 align=center>Search Books</h2>
            <input type="text" name="query" tabindex="1"/>
            <!--<input type="submit" value="Search" tabindex="2"/>-->

            <fieldset>
                <button name="submit" type="submit" value="Search" tabindex="2"
id="contact-submit" >Search</button>
                <button id="contact-submit" type="button" tabindex="3"
onclick="window.location.href='/books-options.php'"> Back </button>
            </fieldset>

        </form>
    </div>
</body>
</html>

<?php
    mysql_connect("localhost", "root", "9667") or die("Error connecting to database:
".mysql_error());
    /*
        localhost - it's location of the mysql server, usually localhost
        root - your username
        third is your password

        if connection fails it will stop loading the page and display an error
    */

    mysql_select_db("kios") or die(mysql_error());
    /* tutorial_search is the name of database we've created */
?>
<!--
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>

```

```

<title>Search results</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>-->
<div class="container">
<form name="form_update" method="post" action="books-search.php">

<fieldset id="contact" class="container" >
<?php
$query = $_GET['query'];
// gets value sent over search form

$min_length = 5;
// you can set minimum length of the query if you want

if(strlen($query) >= $min_length){ // if query length is more or equal minimum length
then

$query = htmlspecialchars($query);
// changes characters used in html to their equivalents, for example: < to &gt;

$query = mysql_real_escape_string($query);
// makes sure nobody uses SQL injection

$raw_results = mysql_query("SELECT * FROM books
WHERE (`title` LIKE '%".$query."%') OR (`author` LIKE '%".$query."%')");
or
die(mysql_error());

// * means that it selects all fields, you can also write: `id`, `title`, `text`
// articles is the name of our table

// '%$query%' is what we're looking for, % means anything, for example if $query is
Hello
    // it will match "hello", "Hello man", "gogohello", if you want exact match use
`title`='$query'
    // or if you want to match just full word so "gogohello" is out use '% $query
%' ... '$query %' ... OR ... '% $query'

$sql2 = mysql_query("SELECT * FROM books WHERE (`title` LIKE '%".$query."%') ")
or die(mysql_error());

```

//SELECT paper title or author FROM journals where like query (searched key)

```
if(mysql_num_rows($raw_results) > 0)
    { // if one or more rows are returned do following
        echo "<select name= 'title'> ";
        echo '<option value="">'.'--- Please Select Title of Books ---'.'</option>';

        while($results = mysql_fetch_array($raw_results))
        {
            // $results = mysql_fetch_array($raw_results) puts data from database into
            array, while it's valid it does the loop

            //echo "<div style ='font:15px/35px Arial,tahoma,sans-
            serif;color:#f6f6f6'<p><h2>".$results['paper_title']. "</h2>".$results['author']. "</p></
            div>";

            echo "<option value='". $results['title']. "'> ".$results['title']
            .'</option>';
        }
        echo '</select>';
    ?>

    <!--<input type="submit" name="submit" value="Submit"/>-->

    <button name="submit" type="submit" value="Submit" id="contact-
    submit">Submit</button>
    </fieldset>
    <?php

}

else{ // if there is no matching rows do following
    echo "No results/ No Journal found!";?
    <fieldset id="container">
    <!-- <button onclick="location.href='/journals.php'" name="submit"
    type="submit" value="Submit" id="contact-submit" > Search</button>-->

    <!-- <button id="contact-submit" type ="sumbit" onClick="parent.location='/
    journals.php'"> New </button>-->
    <button id="contact-submit" type="button" onclick="window.location.href='/books-
    add.php'"> Add Books </button>
    </fieldset>
    <?php
```

```

        }
    }

else{ // if query length is less than minimum
    echo "Minimum length for search is ".$min_length;
}
/*
echo '</select>';
?>
<input type="submit" name="submit" value="Submit"/>
*/
?>

</form>

</div>

<?php

$con=mysqli_connect("localhost","root","9667","kios");
if(mysqli_errno($con))
{
    echo "Can't Connect to mySQL:".mysqli_connect_error();
}

if(isset($_POST['title']))
{
    $title = $_POST['title'];
    $fetch="SELECT title,author,publisher,places,pages,years FROM books WHERE
title = '".$title."'";
    $result = mysqli_query($con,$fetch);
    if(!$result)
    {
        echo "Error:".mysqli_error($con);
    }
    //display the table
    /*
        // In Database Table Journals => paper_title author journal volume numb
pages journal_date
        echo '<table border="1">.<tr>.<td align="center">'. 'From Database'.
'</td>.</tr>';
        //echo '<tr>.<td>.<table border="1">.<tr>.<td>'. 'First Name'. '</
td>.<td>'. 'Last Name'. '</td>.<td>'. 'Gender' . '</td>.<td>'. 'Subject'. '</td>'. '<td>'.
'Hobbies' . '</td>.</tr>';
    */
}

```

```

echo '<tr>.<td>.<table border="1">.<tr>.<td>.'Paper Title'.</td>.<td>.'Author'.</td>.<td>.'Journal'.</td>.<td>.'Volume'.</td>.<td>.'Number'.</td>.<td>.'Pages'.</td>.<td>.'Journal Date'.</td>.</tr>';

//while($data = mysqli_fetch_row($fetch))
while($data=mysqli_fetch_row($result))
{
    echo ("<tr><td>$data[0]</td><td>$data[1]</td><td>$data[2]</td><td>$data[3]</td><td>$data[4]</td><td>$data[5]</td><td>$data[6]</td></tr>");
}
echo '</table>.</td>.</tr>.</table>'*/


?>
<div class="container">
<!-- Form To Display Record From Database In Inputs --&gt;
&lt;form id="contact" name="form" method="POST" action = "books-
display.php"&gt;
    &lt;h2 align=center&gt;Books Form Updated&lt;/h2&gt;
    &lt;!-- In Database Table Journals =&gt; paper_title author journal volume
numb pages journal_date --&gt;
    &lt;?php while ($data2=mysqli_fetch_row($result))
    {
        ?&gt;
        Title: &lt;input type = "text" name="Title" value = "&lt;?php echo $data2[0]?
&gt;"/&gt; &lt;br&gt;
        Author: &lt;input type = "text" name="Author" value = "&lt;?php echo
$data2[1]?&gt;"/&gt; &lt;br&gt;
        Publisher: &lt;input type = "text" name="Publisher" value = "&lt;?php echo
$data2[2]?&gt;"/&gt; &lt;br&gt;
        Places: &lt;input type = "text" name="Places" value = "&lt;?php echo
$data2[3]?&gt;"/&gt; &lt;br&gt;
        Pages: &lt;input type = "text" name="Pages" value = "&lt;?php echo
$data2[4]?&gt;"/&gt; &lt;br&gt;
        Years: &lt;input type = "text" name="Years" value = "&lt;?php echo
$data2[5]?&gt;"/&gt; &lt;br&gt;&lt;br&gt;
        &lt;?php
    }
    ?&gt;
    &lt;input type="hidden" name="keyfield" value="&lt;?php echo $search?&gt;"&gt;
    &lt;!--&lt;button type="submit" value="submit" id="contact-submit"&gt;--&gt;
    &lt;fieldset&gt;
        &lt;button name="submit" type="submit" value="Submit" id="contact-
submit" &gt;Edit/Update&lt;/button&gt;
</pre>

```

```

        </fieldset>
    </form>
</div>
<?php
}
?>
```

GENERAL-SEARCH.PHP

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<style>
@import url(https://fonts.googleapis.com/css?family=Roboto:400,300,600,400italic);
margin: 0;
padding: 0;
box-sizing: border-box;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
-webkit-font-smoothing: antialiased;
-moz-font-smoothing: antialiased;
-o-font-smoothing: antialiased;
font-smoothing: antialiased;
text-rendering: optimizeLegibility;
}

body {
font-family: "Roboto", Helvetica, Arial, sans-serif;
font-weight: 100;
font-size: 14px;
line-height: 30px;
color: #777;
background: #4CAF50;
}

.container {
max-width: 800px;
width: 100%;
margin: 0 auto;
position: relative;
}

#contact input[type="text"],
#contact input[type="email"],
```

```
#contact input[type="tel"],  
#contact input[type="url"],  
#contact textarea,  
#contact button[type="submit"] {  
    font: 400 12px/16px "Roboto", Helvetica, Arial, sans-serif;  
}  
  
#contact {  
    background: #F9F9F9;  
    padding: 25px;  
    margin: 150px 0;  
    box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);  
}  
  
#contact h3 {  
    display: block;  
    font-size: 30px;  
    font-weight: 300;  
    margin-bottom: 10px;  
}  
  
#contact h4 {  
    margin: 5px 0 15px;  
    display: block;  
    font-size: 13px;  
    font-weight: 400;  
}  
  
fieldset {  
    border: medium none !important;  
    margin: 0 0 10px;  
    min-width: 100%;  
    padding: 0;  
    width: 100%;  
}  
  
#contact input[type="text"],  
#contact input[type="email"],  
#contact input[type="tel"],  
#contact input[type="url"],  
#contact textarea {  
    width: 100%;  
    border: 1px solid #ccc;  
    background: #FFF;
```

```
margin: 0 0 5px;
padding: 10px;
}

#contact input[type="text"]:hover,
#contact input[type="email"]:hover,
#contact input[type="tel"]:hover,
#contact input[type="url"]:hover,
#contact textarea:hover {
    -webkit-transition: border-color 0.3s ease-in-out;
    -moz-transition: border-color 0.3s ease-in-out;
    transition: border-color 0.3s ease-in-out;
    border: 1px solid #aaa;
}

#contact textarea {
    height: 100px;
    max-width: 100%;
    resize: none;
}

#contact button[type="submit"] {
    cursor: pointer;
    width: 100%;
    border: none;
    background: #4CAF50;
    color: #FFF;
    margin: 0 0 5px;
    padding: 10px;
    font-size: 15px;
}

#contact button[type="submit"]:hover {
    background: #43A047;
    -webkit-transition: background 0.3s ease-in-out;
    -moz-transition: background 0.3s ease-in-out;
    transition: background-color 0.3s ease-in-out;
}

#contact button[type="submit"]:active {
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);
}
```

```
/*NEW CODE HERE*/  
  
#contact button[type="button"] {  
    cursor: pointer;  
    width: 100%;  
    border: none;  
    background: #4CAF50;  
    color: #FFF;  
    margin: 0 0 5px;  
    padding: 10px;  
    font-size: 15px;  
}  
  
#contact button[type="button"]:hover {  
    background: #43A047;  
    -webkit-transition: background 0.3s ease-in-out;  
    -moz-transition: background 0.3s ease-in-out;  
    transition: background-color 0.3s ease-in-out;  
}  
  
#contact button[type="button"]:active {  
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.5);  
}  
/*END OF NEW CODE*/  
.copyright {  
    text-align: center;  
}  
  
#contact input:focus,  
#contact textarea:focus {  
    outline: 0;  
    border: 1px solid #aaa;  
}  
  
::-webkit-input-placeholder {  
    color: #888;  
}  
::-moz-placeholder {  
    color: #888;  
}  
::placeholder {  
    color: #888;
```

}

```
: -ms-input-placeholder {  
    color: #888;  
}  
</style>
```

<!--Report Options

1. Any combination of publication type:

Books
Journals
Book Chapters
Conferences
Patents
Other
Select All

2. Any combination of author(s):

1. All in database
2. Single author(s)
3. List of author(s) common publication(s) // search solution for "OR"
4. List of author(s) combined publication(s) // search solution for "AND"

3. Year of Publication

1. All
2. Specific year
3. Range

Buttons = [Submit]

PREVIEW TABLE

Buttons = [Save as PDF - Export to Excel - Print]

Title

Web-based Database System Publications

PRESENT IN COMMA SEPARATED VALUES

(Excel must be tables)

in PDF in CSV format

[TABLE OF SEARCHED RETRIEVED FROM DATABASE]-->

<!--

RECAP: So the sql should be present everything is csv format, not tables, and the sql query should be only present the search values. So for example if we search in the tables of journals for authors and the year, that line should be presented. THE ENTIRE ROW ONLY.

```
<!--
<?php
/*
if(isset($_POST)) {
    $sql = "SELECT * FROM `table` "
    if($_POST['price_min'] != 'doesntmatter') {
        $sql .= "WHERE `price` > '". $_POST['price_min']. "' ";
    }
    if($_POST['price_max'] != 'doesntmatter') {
        $sql .= "WHERE `price` < '". $_POST['price_max']. "' ";
    }
}
?>-->
```

```
<head>
    <title>Search</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
    <div class="container">
        <form id="contact" action="general-search.php" method="post">

            <h1 align=center>Compile Report</h1>
            <h2 align=left> Report Options:</h2>
            <h2 align=left> 1.Select any combination of publication type:</h2>

            <input type="checkbox" name="check_list[]" value="books"><label>Books</label><br>
            <input type="checkbox" name="check_list[]" value="journals"><label>Journals</label><br>
            <input type="checkbox" name="check_list[]" value="books_chapters"><label>Book Chapters</label><br>
            <input type="checkbox" name="check_list[]" value="conferences"><label>Conferences</label><br>
```

```

<input type="checkbox" name="check_list[]" value="patents"><label>Patents</label><br>
<input type="checkbox" name="check_list[]" value="other"><label>Other</label><br>
<th> <input type="checkbox" onchange="checkAll(this)" />Select All</th>
<br>
<h2 align=left> 2.Select any combination of author(s):</h2>
<?php $conn=new mysqli("localhost","root","9667","kios");

//CORRECT
$rows=$conn->query("SELECT author FROM books UNION SELECT author FROM
journals UNION SELECT author FROM book_chapters UNION SELECT author FROM
other UNION SELECT author FROM patents");
//try $rows=$conn->query("SELECT * FROM books LEFT JOIN book_chapters USING
(years)";

//$rows=$conn->query("SELECT books.title ,book_chapters.years FROM books LEFT
JOIN book_chapters ON books.years >= book_chapters.years WHERE books.years =
'2009'";
/*
SELECT Customers.CustomerName, Orders.OrderID
FROM Customers
LEFT JOIN Orders
ON Customers.CustomerID=Orders.CustomerID
WHERE Customers.CustomerID ='2';*/

```

```
while(list($authors)=$rows->fetch_row()) {
```

```
?>
```

```

<input type="checkbox" name="form[]" value=" "> <?php echo htmlspecialchars
("$authors") ?>" > <label> <?php echo "$authors" ?></label><br>

<?php
}
//echo "</table>";
?>
<th> <input type="checkbox" onchange="checkAll2(this)" />Select All</th>
<br>
<h2 align=left> 3.Select any combination of year(s) range:</h2>

<fieldset>
<span> <?php echo 'Date From:';?> </span>

```

```

<input placeholder="Date eg.Y-M-D-> 2001-09-25" type="text" name ="DateFrom">

<span> <?php echo 'Date To:';?> </span>

<input placeholder="Date eg.Y-M-D-> 2012-10-26" type="text" name ="DateTo">

</fieldset>

<br>

<fieldset>
    <button name="submit" type="submit" value="Search" id="contact-submit">
        Search </button>
    <button id="contact-submit" type="button" onclick="window.location.href='/
index.php'"> Back </button><br>
</fieldset>
<!---- Including PHP Script ----->

<?php
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>

<fieldset class = "container" >
<?php
if(isset($_POST['submit'])){

/*
if(!empty($_POST['DateFrom']) AND !empty($_POST['DateTo'])) {

    $conn=new mysqli("localhost", "root", "9667", "kios");
    $datefrom = test_input($_POST['DateFrom']);
    $dateto = test_input($_POST['DateTo']);
    $rows = $conn->query("SELECT journal_date FROM journals WHERE journal_date >=
'$datefrom' AND journal_date <= '$dateto'");


```

```

while(list($date)=$rows->fetch_row()) {
echo "<p>" . $date . "</p>";

}

}

else {

echo "<b>Please complete all dates above in order to search.</b>";
}

if(!empty($_POST['check_list'])) {
// Counting number of checked checkboxes.
$checked_count = count($_POST['check_list']);
echo "You have selected following ".$checked_count." option(s): <br/>";
// Loop to store and display values of individual checked checkbox.
foreach($_POST['check_list'] as $selected)
{

echo "<p>".$selected . "</p>";

}

else{
echo "<b>Please Select At least One Option.</b>";
}

$aDoor = $_POST['form'];
if(empty($aDoor))
{
echo("You didn't select any author(s). ");
}
else
{
$N = count($aDoor);
echo("You selected $N author(s): ");
for($i=0; $i < $N; $i++)
{
echo "<p>" . $aDoor[$i] . "</p>";
}
}

*/
//TRYING

```

```

if(!empty($_POST['DateFrom']) AND !empty($_POST['DateTo']) AND !
empty($_POST['check_list']) AND !empty($_POST['form']))
{
    $conn=new mysqli("localhost", "root", "9667", "kios");
    $datefrom = test_input($_POST['DateFrom']);
    $dateto = test_input($_POST['DateTo']);
    $aDoor = $_POST['form'];
    $loop = $_POST['check_list'];
    $checked_count_tables = count($_POST['check_list']); // length
of checklist
    $checked_count_authors = count($_POST['form']); // length of
form for authors

/*
//PRINTS SELECTED VALUES OF TABLES
foreach($_POST['check_list'] as $selected)
{
    echo "<p>".$selected."</p>";
}

$checked_count_authors = count($aDoor);
echo("You selected $checked_count_authors author(s): ");

//PRINT SELECTED VALUES OF AUTHORS
for($i=0; $i < $checked_count_authors; $i++)
{
    echo "<p> . $aDoor[$i] . "</p>";
}

//PRINT SELECTED VALUES OF DATES
echo " $datefrom";
echo " ";
echo "$dateto";
*/


//(DOESN'T WORK) QUERY TO RETURN AUTHORS FROM ALL TABLES
/* $rows = $conn ->query (" SELECT authors FROM ";
for($i=0; $i <$checked_count_tables; $i++)
{
    "$loop[$i]";
} " ");
while(list($query)=$rows->fetch_row()) {
    echo "<p> " . $query . "</p>";
}

```

```

    }*/
    echo "<p> SEARCH RESULTS FROM SELECTED
CHECKBOXES</p>";
    echo "<p>";

    // $rows=$conn->query("SELECT author FROM books UNION
SELECT author FROM journals UNION SELECT author FROM book_chapters UNION
SELECT author FROM other UNION SELECT author FROM patents");
    // $rows=$conn->query("SELECT author AND years WHERE
'2009' FROM books UNION SELECT author AND years WHERE '1993' FROM
book_chapters");
    // $rows=$conn->query("SELECT * FROM table WHERE
YEAR(date)=2012
    // "SELECT journal_date FROM journals WHERE journal_date >=
'$datefrom' AND journal_date <= '$dateto'");

    //test
    // $rows=$conn->query("SELECT title,author,years FROM books
WHERE years >= '1999' and years <= '2015'  ");

    //title author publisher places pages years
/*
        while(list($authors)=$rows->fetch_row()) {
            echo "<p> ".$authors."<p> ";
            $con=mysqli_connect("localhost","root","9667","kios");
            if(mysqli_errno($con))
            {
                echo "Can't Connect to mySQL:".mysqli_connect_error();
            }
            $fetch="SELECT
title,author,publisher,places,pages,years FROM books WHERE years >= '1999' and years
<= '2015' ";
            $result = mysqli_query($con,$fetch);
            if(!$result)
            {
                echo "Error: ".(mysqli_error($con));
            }
        }
    
```

```

echo '<table border="1">'. '<tr>'. '<td
align="center">'. 'From Database'. '</td>'. '</tr>';
echo '<tr>'. '<td>'. '<table
border="1">'. '<tr>'. '<td>'. 'Title'. '</td>'. '<td>'. 'Author'. '</td>'. '<td>'. 'Publisher' . '<
td>'. '<td>'. 'Places' . '</td>'. '<td>'. 'Pages' . '</td>'. '<td>'. 'Years' . '</td>'. '</tr>';

while ($datas=mysqli_fetch_row($result))
{
    ?>

    <tr>
        <td><?php echo $datas[0]; ?></td>
        <td><?php echo $datas[1]; ?></td>
        <td><?php echo $datas[2]; ?></td>
        <td><?php echo $datas[3]; ?></td>
        <td><?php echo $datas[4]; ?></td>
        <td><?php echo $datas[5]; ?></td>

    </tr>
    <?php
}
echo '</table>'. '</td>'. '</tr>'. '</table>';
echo "<p>";

    ?>

<fieldset>
    <button id="contact-submit" type="button"
onclick="window.location.href='/index.php'"> Back to Main Page</button>
    </fieldset>
<fieldset>
    <button id="contact-submit" type="button"
onclick="window.location.href='/journals-export.php'"> Export in CSV</button>
    </fieldset>
<fieldset>
    <button id="contact-submit" type="button"
onclick="window.location.href='/journals-export.php'"> Export in PDF</button>
    </fieldset>
<fieldset>
    <button id="contact-submit" type="button">
Print</button>
    </fieldset>

```

<?php

```
//EDUCATIONAL
    // $rows = $conn->query("SELECT journal_date FROM journals
WHERE journal_date >= '$datefrom' AND journal_date <= '$dateto'");
    //while(list($date)=$rows->fetch_row()) {
    // echo "<p>" . $date . "</p>";
    /*Use an IN query, which will grab all of the results in a single
query:
    SELECT * FROM people WHERE town IN('LA', 'London',
'Paris')*/
}
```

```
}
```

```
?>
```

```
</fieldset>
</form>
```

```
<script>
function checkAll(ele) {
    var checkboxes = document.getElementsByName('input');
```

```

if (ele.checked) {
    for (var i = 0; i < checkboxes.length; i++) {
        if (checkboxes[i].type == 'checkbox' && checkboxes[i].name =='check_list[]' ) {
            checkboxes[i].checked = true;
        }
    }
} else {
    for (var i = 0; i < checkboxes.length; i++) {
        console.log(i)
        if (checkboxes[i].type == 'checkbox' && checkboxes[i].name =='check_list[]' ) {
            checkboxes[i].checked = false;
        }
    }
}

function checkAll2(ele) {
    var checkboxes = document.getElementsByTagName('input');
    if (ele.checked) {
        for (var i = 0; i < checkboxes.length; i++) {
            if (checkboxes[i].type == 'checkbox' && checkboxes[i].name =='form[]' ) {
                checkboxes[i].checked = true;
            }
        }
    } else {
        for (var i = 0; i < checkboxes.length; i++) {
            console.log(i)
            if (checkboxes[i].type == 'checkbox' && checkboxes[i].name =='form[]' ) {
                checkboxes[i].checked = false;
            }
        }
    }
}

</script>
</div>
</body>
</html>

```

JAVASCRIPT

JQuery.js

```
/*
 * jQuery JavaScript Library v1.10.2
 * http://jquery.com/
 *
 * Includes Sizzle.js
 * http://sizzlejs.com/
 *
 * Copyright 2005, 2013 jQuery Foundation, Inc. and other contributors
 * Released under the MIT license
 * http://jquery.org/license
 *
 * Date: 2013-07-03T13:48Z
 */
(function( window, undefined ) {

// Can't do this because several apps including ASP.NET trace
// the stack via arguments.caller.callee and Firefox dies if
// you try to trace through "use strict" call chains. (#13335)
// Support: Firefox 18+
// "use strict";
var
    // The deferred used on DOM ready
    readyList,
    // A central reference to the root jQuery(document)
    rootjQuery,
    // Support: IE<10
    // For `typeof xmlNode.method` instead of `xmlNode.method !== undefined`
    core_undefined = typeof undefined,
    // Use the correct document accordingly with window argument (sandbox)
    location = window.location,
    document = window.document,
    docElem = document.documentElement,
    // Map over jQuery in case of overwrite
    _jQuery = window.jQuery,
    // Map over the $ in case of overwrite
})()
```

```

_ = window._,
// [[Class]] -> type pairs
class2type = {},

// List of deleted data cache ids, so we can reuse them
core_deletedIds = [],

core_version = "1.10.2",

// Save a reference to some core methods
core_concat = core_deletedIds.concat,
core_push = core_deletedIds.push,
core_slice = core_deletedIds.slice,
core_indexOf = core_deletedIds.indexOf,
core_toString = class2type.toString,
core_hasOwn = class2type.hasOwnProperty,
core_trim = core_version.trim,

// Define a local copy of jQuery
jQuery = function( selector, context ) {
    // The jQuery object is actually just the init constructor 'enhanced'
    // with several extra functions attached. This allows you to chain
    // methods like .filter() which returns you a new wrapped jqObject
    // rather than the DOM element. Thus it is best to chain these
    // methods in this order: query, filter, map, reduce, etc.
    return new jQuery.fn.init( selector, context, rootjQuery );
},

// Used for matching numbers
core_pnum = /[+-]?(?:\d*\.\d|)\d+(?:[eE][+-]?\d+|)/.source,

// Used for splitting on whitespace
core_rnotwhite = /\S+/g,

// Make sure we trim BOM and NBSP (here's looking at you, Safari 5.0 and IE)
rtrim = /[\s\uFEFF\xA0]+|[^\s\uFEFF\xA0]+$/g,

// A simple way to check for HTML strings
// Prioritize #id over <tag> to avoid XSS via location.hash (#9521)
// Strict HTML recognition (#11290: must start with <)
rquickExpr = /^(?:\s*(?<[\w\W]+>)[^>]*#([\w-]*))$/,

// Match a standalone tag
rsingleTag = /^<(\w+)\s*\V?>(?:<\V1>)$/,

// JSON RegExp
rvalidchars = /^[\\],:{}\\s]*$/,
rvalidbraces = /(?:^|:|,)(?:\\s*\\|)+/g,

```

```

rvalidescape = /\\"(?:\"\\bfnrt]lu[\\da-fA-F]{4})/g,
rvalidtokens = /[^\"\\r\\n]*"ltruefalse|null|-?(?:\\d+\\.|)\\d+(?:[eE][+-]?\\d+)/g,

// Matches dashed string for camelizing
rmsPrefix = /^-ms-/,
rdashAlpha = /-(\\da-z))/gi,

// Used by jQuery.camelCase as callback to replace()
fcamelCase = function( all, letter ) {
    return letter.toUpperCase();
},

// The ready event handler
completed = function( event ) {

    // readyState === "complete" is good enough for us to call the dom ready
in oldIE
    if ( document.addEventListener || event.type === "load" ||
document.readyState === "complete" ) {
        detach();
        jQuery.ready();
    }
},
// Clean-up method for dom ready events
detach = function() {
    if ( document.addEventListener ) {
        document.removeEventListener( "DOMContentLoaded", completed,
false );
        window.removeEventListener( "load", completed, false );
    } else {
        document.detachEvent( "onreadystatechange", completed );
        window.detachEvent( "onload", completed );
    }
};

jQuery.fn = jQuery.prototype = {
    // The current version of jQuery being used
    jquery: core_version,

    constructor: jQuery,
    init: function( selector, context, rootjQuery ) {
        var match, elem;

        // HANDLE: $(""), $(null), $(undefined), $(false)

```

```

if ( !selector ) {
    return this;
}

// Handle HTML strings
if ( typeof selector === "string" ) {
    if ( selector.charAt(0) === "<" && selector.charAt( selector.length - 1 )
===== ">" && selector.length >= 3 ) {
        // Assume that strings that start and end with <> are HTML
        and skip the regex check
        match = [ null, selector, null ];
    }

} else {
    match = rquickExpr.exec( selector );
}

// Match html or make sure no context is specified for #id
if ( match && (match[1] || !context) ) {

    // HANDLE: $(html) -> $(array)
    if ( match[1] ) {
        context = context instanceof jQuery ? context[0] :
context;

        // scripts is true for back-compat
        jQuery.merge( this, jQuery.parseHTML(
            match[1],
            context && context.nodeType ?
context.ownerDocument || context : document,
            true
        ));

        // HANDLE: $(html, props)
        if ( rsingleTag.test( match[1] ) &&
jQuery.isPlainObject( context ) ) {
            for ( match in context ) {
                // Properties of context are called as
                methods if possible
                if ( jQueryisFunction( this[ match ] ) ) {
                    this[ match ]( context[ match ] );

                    // ...and otherwise set as attributes
                } else {
                    this.attr( match, context[ match ] );
                }
            }
        }
    }
}

```

```

        }

    }

    return this;

    // HANDLE: $(#id)
} else {
    elem = document.getElementById( match[2] );

    // Check parentNode to catch when BlackBerry 4.6
returns

    // nodes that are no longer in the document #6963
    if ( elem && elem.parentNode ) {
        // Handle the case where IE and Opera return
items

        // by name instead of ID
        if ( elem.id !== match[2] ) {
            return rootjQuery.find( selector );
        }

        // Otherwise, we inject the element directly into
the jQuery object
this.length = 1;
this[0] = elem;
}

    this.context = document;
    this.selector = selector;
    return this;
}

// HANDLE: $(expr, $(...))
} else if ( !context || context.jquery ) {
    return ( context || rootjQuery ).find( selector );

    // HANDLE: $(expr, context)
    // (which is just equivalent to: $(context).find(expr)
} else {
    return this.constructor( context ).find( selector );
}

// HANDLE: $(DOMElement)
} else if ( selector.nodeType ) {
    this.context = this[0] = selector;
    this.length = 1;
}

```

```

        return this;

    // HANDLE: $(function)
    // Shortcut for document ready
} else if ( jQuery.isFunction( selector ) ) {
    return rootjQuery.ready( selector );
}

if ( selector.selector !== undefined ) {
    this.selector = selector.selector;
    this.context = selector.context;
}

return jQuery.makeArray( selector, this );
},

// Start with an empty selector
selector: " ",

// The default length of a jQuery object is 0
length: 0,

toArray: function() {
    return core_slice.call( this );
},

// Get the Nth element in the matched element set OR
// Get the whole matched element set as a clean array
get: function( num ) {
    return num == null ?

        // Return a 'clean' array
        this.toArray() :

        // Return just the object
        ( num < 0 ? this[ this.length + num ] : this[ num ] );
},

// Take an array of elements and push it onto the stack
// (returning the new matched element set)
pushStack: function( elems ) {

    // Build a new jQuery matched element set
    var ret = jQuery.merge( this.constructor(), elems );
}

```

```
// Add the old object onto the stack (as a reference)
ret.prevObject = this;
ret.context = this.context;

// Return the newly-formed element set
return ret;
},

// Execute a callback for every element in the matched set.
// (You can seed the arguments with an array of args, but this is
// only used internally.)
each: function( callback, args ) {
    return jQuery.each( this, callback, args );
},

ready: function( fn ) {
    // Add the callback
    jQuery.ready.promise().done( fn );

    return this;
},

slice: function() {
    return this.pushStack( core_slice.apply( this, arguments ) );
},

first: function() {
    return this.eq( 0 );
},

last: function() {
    return this.eq( -1 );
},

eq: function( i ) {
    var len = this.length,
        j = +i + ( i < 0 ? len : 0 );
    return this.pushStack( j >= 0 && j < len ? [ this[j] ] : [] );
},

map: function( callback ) {
    return this.pushStack( jQuery.map(this, function( elem, i ) {
        return callback.call( elem, i, elem );
    }));
},
```

```

end: function() {
    return this.prevObject || this.constructor(null);
},
// For internal use only.
// Behaves like an Array's method, not like a jQuery method.
push: core_push,
sort: [].sort,
splice: [].splice
};

// Give the init function the jQuery prototype for later instantiation
jQuery.fn.init.prototype = jQuery.fn;

jQuery.extend = jQuery.fn.extend = function() {
    var src, copyIsArray, copy, name, options, clone,
        target = arguments[0] || {},
        i = 1,
        length = arguments.length,
        deep = false;

    // Handle a deep copy situation
    if ( typeof target === "boolean" ) {
        deep = target;
        target = arguments[1] || {};
        // skip the boolean and the target
        i = 2;
    }

    // Handle case when target is a string or something (possible in deep copy)
    if ( typeof target !== "object" && !jQueryisFunction(target) ) {
        target = {};
    }

    // extend jQuery itself if only one argument is passed
    if ( length === i ) {
        target = this;
        --i;
    }

    for ( ; i < length; i++ ) {
        // Only deal with non-null/undefined values
        if ( (options = arguments[ i ]) != null ) {
            // Extend the base object

```

```

        for ( name in options ) {
            src = target[ name ];
            copy = options[ name ];

            // Prevent never-ending loop
            if ( target === copy ) {
                continue;
            }

            // Recurse if we're merging plain objects or arrays
            if ( deep && copy && ( jQuery.isPlainObject(copy) ||
(copyIsArray = jQuery.isArray(copy)) ) ) {
                if ( copyIsArray ) {
                    copyIsArray = false;
                    clone = src && jQuery.isArray(src) ? src : [];

                } else {
                    clone = src && jQuery.isPlainObject(src) ? src : {};
                }

                // Never move original objects, clone them
                target[ name ] = jQuery.extend( deep, clone, copy );
            }

            // Don't bring in undefined values
            } else if ( copy !== undefined ) {
                target[ name ] = copy;
            }
        }
    }

    // Return the modified object
    return target;
};

jQuery.extend({
    // Unique for each copy of jQuery on the page
    // Non-digits removed to match rinlinejQuery
    expando: "jQuery" + ( core_version + Math.random() ).replace( /\D/g, "" ),

    noConflict: function( deep ) {
        if ( window.$ === jQuery ) {
            window.$ = _$;
        }
    }
});

```

```
if ( deep && window.jQuery === jQuery ) {
    window.jQuery = _jQuery;
}

return jQuery;
},

// Is the DOM ready to be used? Set to true once it occurs.
isReady: false,

// A counter to track how many items to wait for before
// the ready event fires. See #6781
readyWait: 1,

// Hold (or release) the ready event
holdReady: function( hold ) {
    if ( hold ) {
        jQuery.readyWait++;
    } else {
        jQuery.ready( true );
    }
},

// Handle when the DOM is ready
ready: function( wait ) {

    // Abort if there are pending holds or we're already ready
    if ( wait === true ? --jQuery.readyWait : jQuery.isReady ) {
        return;
    }

    // Make sure body exists, at least, in case IE gets a little overzealous (ticket
    #5443).
    if ( !document.body ) {
        return setTimeout( jQuery.ready );
    }

    // Remember that the DOM is ready
    jQuery.isReady = true;

    // If a normal DOM Ready event fired, decrement, and wait if need be
    if ( wait !== true && --jQuery.readyWait > 0 ) {
        return;
    }
}
```

```

// If there are functions bound, to execute
readyList.resolveWith( document, [ jQuery ] );

// Trigger any bound ready events
if ( jQuery.fn.trigger ) {
    jQuery( document ).trigger("ready").off("ready");
}
},

// See test/unit/core.js for details concerning isFunction.
// Since version 1.3, DOM methods and functions like alert
// aren't supported. They return false on IE (#2968).
isFunction: function( obj ) {
    return jQuery.type(obj) === "function";
},

isArray: Array.isArray || function( obj ) {
    return jQuery.type(obj) === "array";
},
}

isWindow: function( obj ) {
    /* jshint eqeqeq: false */
    return obj != null && obj == obj.window;
},
}

isNumeric: function( obj ) {
    return !isNaN( parseFloat(obj) ) && isFinite( obj );
},
}

type: function( obj ) {
    if ( obj == null ) {
        return String( obj );
    }
    return typeof obj === "object" || typeof obj === "function" ?
        class2type[ core_toString.call(obj) ] || "object" :
        typeof obj;
},
}

isPlainObject: function( obj ) {
    var key;

    // Must be an Object.
    // Because of IE, we also have to check the presence of the constructor
    property.
}

```

```

// Make sure that DOM nodes and window objects don't pass through, as
well
if ( !obj || jQuery.type(obj) !== "object" || obj.nodeType ||
jQuery.isWindow( obj ) ) {
    return false;
}

try {
    // Not own constructor property must be Object
    if ( obj.constructor &&
        !core_hasOwn.call(obj, "constructor") &&
        !core_hasOwn.call(obj.constructor.prototype,
"isPrototypeOf") ) {
        return false;
    }
} catch ( e ) {
    // IE8,9 Will throw exceptions on certain host objects #9897
    return false;
}

// Support: IE<9
// Handle iteration over inherited properties before own properties.
if ( jQuery.support.ownLast ) {
    for ( key in obj ) {
        return core_hasOwn.call( obj, key );
    }
}

// Own properties are enumerated firstly, so to speed up,
// if last one is own, then all properties are own.
for ( key in obj ) {

    return key === undefined || core_hasOwn.call( obj, key );
},

isEmptyObject: function( obj ) {
    var name;
    for ( name in obj ) {
        return false;
    }
    return true;
},

error: function( msg ) {
    throw new Error( msg );
}

```

```

    },

    // data: string of html
    // context (optional): If specified, the fragment will be created in this context,
    defaults to document
    // keepScripts (optional): If true, will include scripts passed in the html string
    parseHTML: function( data, context, keepScripts ) {
        if ( !data || typeof data !== "string" ) {
            return null;
        }
        if ( typeof context === "boolean" ) {
            keepScripts = context;
            context = false;
        }
        context = context || document;

        var parsed = rsingleTag.exec( data ),
            scripts = !keepScripts && [];

        // Single tag
        if ( parsed ) {
            return [ context.createElement( parsed[1] ) ];
        }

        parsed = jQuery.buildFragment( [ data ], context, scripts );
        if ( scripts ) {
            jQuery( scripts ).remove();
        }
        return jQuery.merge( [], parsed.childNodes );
    },

    parseJSON: function( data ) {
        // Attempt to parse using the native JSON parser first
        if ( window.JSON && window.JSON.parse ) {
            return window.JSON.parse( data );
        }

        if ( data === null ) {
            return data;
        }

        if ( typeof data === "string" ) {

            // Make sure leading/trailing whitespace is removed (IE can't handle
            it)
        }
    }
}

```

```

        data = jQuery.trim( data );

        if ( data ) {
            // Make sure the incoming data is actual JSON
            // Logic borrowed from http://json.org/json2.js
            if ( rvalidchars.test( data.replace( rvalidescape, "@" )
                .replace( rvalidtokens, "]" )
                .replace( rvalidbraces, "" ) ) {

                return ( new Function( "return " + data ) )();
            }
        }
    }

    jQuery.error( "Invalid JSON: " + data );
},

// Cross-browser xml parsing
parseXML: function( data ) {
    var xml, tmp;
    if ( !data || typeof data !== "string" ) {
        return null;
    }
    try {
        if ( window.DOMParser ) { // Standard
            tmp = new DOMParser();
            xml = tmp.parseFromString( data , "text/xml" );
        } else { // IE
            xml = new ActiveXObject( "Microsoft.XMLDOM" );
            xml.async = "false";
            xml.loadXML( data );
        }
    } catch( e ) {
        xml = undefined;
    }
    if ( !xml || !xml.documentElement ||
        xml.getElementsByTagName( "parsererror" ).length ) {
        jQuery.error( "Invalid XML: " + data );
    }
    return xml;
},
noop: function() {},  

// Evaluates a script in a global context

```

```

// Workarounds based on findings by Jim Driscoll
// http://weblogs.java.net/blog/driscoll/archive/2009/09/08/eval-javascript-global-
context
globalEval: function( data ) {
    if ( data && jQuery.trim( data ) ) {
        // We use execScript on Internet Explorer
        // We use an anonymous function so that context is window
        // rather than jQuery in Firefox
        ( window.execScript || function( data ) {
            window[ "eval" ].call( window, data );
        } )( data );
    }
},
// Convert dashed to camelCase; used by the css and data modules
// Microsoft forgot to hump their vendor prefix (#9572)
camelCase: function( string ) {
    return string.replace( rmsPrefix, "ms-" ).replace( rdashAlpha, fcamelCase );
},
nodeName: function( elem, name ) {
    return elem.nodeName && elem.nodeName.toLowerCase() ===
name.toLowerCase();
},
// args is for internal usage only
each: function( obj, callback, args ) {
    var value,
        i = 0,
        length = obj.length,
        isArray = isArraylike( obj );

    if ( args ) {
        if ( isArray ) {
            for ( ; i < length; i++ ) {
                value = callback.apply( obj[ i ], args );

                if ( value === false ) {
                    break;
                }
            }
        } else {
            for ( i in obj ) {
                value = callback.apply( obj[ i ], args );
            }
        }
    }
}

```

```

        if ( value === false ) {
            break;
        }
    }

// A special, fast, case for the most common use of each
} else {
    if ( isArray ) {
        for ( ; i < length; i++ ) {
            value = callback.call( obj[ i ], i, obj[ i ] );

            if ( value === false ) {
                break;
            }
        }
    } else {
        for ( i in obj ) {
            value = callback.call( obj[ i ], i, obj[ i ] );

            if ( value === false ) {
                break;
            }
        }
    }
}

return obj;
},

// Use native String.trim function wherever possible
trim: core_trim && !core_trim.call("\uFEFF\xA0") ?
    function( text ) {
        return text == null ?
            "" :
            core_trim.call( text );
} :

// Otherwise use our own trimming functionality
function( text ) {
    return text == null ?
        "" :
        ( text + "" ).replace( rtrim, "" );
},

```

```

// results is for internal usage only
makeArray: function( arr, results ) {
    var ret = results || [];

    if ( arr != null ) {
        if ( isArraylike( Object(arr) ) ) {
            jQuery.merge( ret,
                typeof arr === "string" ?
                    [ arr ] : arr
            );
        } else {
            core_push.call( ret, arr );
        }
    }

    return ret;
},

inArray: function( elem, arr, i ) {
    var len;

    if ( arr ) {
        if ( core_indexOf ) {
            return core_indexOf.call( arr, elem, i );
        }

        len = arr.length;
        i = i ? i < 0 ? Math.max( 0, len + i ) : i : 0;

        for ( ; i < len; i++ ) {
            // Skip accessing in sparse arrays
            if ( i in arr && arr[ i ] === elem ) {
                return i;
            }
        }
    }

    return -1;
},

merge: function( first, second ) {
    var l = second.length,
        i = first.length,
        j = 0;
}

```

```

if ( typeof l === "number" ) {
    for ( ; j < l; j++ ){
        first[ i++ ] = second[ j ];
    }
} else {
    while ( second[j] !== undefined ){
        first[ i++ ] = second[ j++ ];
    }
}

first.length = i;

return first;
},


grep: function( elems, callback, inv ) {
    var retVal,
        ret = [],
        i = 0,
        length = elems.length;
    inv = !!inv;

    // Go through the array, only saving the items
    // that pass the validator function
    for ( ; i < length; i++ ){
        retVal = !!callback( elems[ i ], i );
        if ( inv !== retVal ) {
            ret.push( elems[ i ] );
        }
    }

    return ret;
},


// arg is for internal usage only
map: function( elems, callback, arg ) {
    var value,
        i = 0,
        length = elems.length,
        isArray = isArraylike( elems ),
        ret = [];

    // Go through the array, translating each of the items to their
    if ( isArray ) {
        for ( ; i < length; i++ ){

```

```

        value = callback( elems[ i ], i, arg );

        if ( value != null ) {
            ret[ ret.length ] = value;
        }
    }

// Go through every key on the object,
} else {
    for ( i in elems ) {
        value = callback( elems[ i ], i, arg );

        if ( value != null ) {
            ret[ ret.length ] = value;
        }
    }
}

// Flatten any nested arrays
return core_concat.apply( [], ret );
},

// A global GUID counter for objects
guid: 1,

// Bind a function to a context, optionally partially applying any
// arguments.
proxy: function( fn, context ) {
    var args, proxy, tmp;

    if ( typeof context === "string" ) {
        tmp = fn[ context ];
        context = fn;
        fn = tmp;
    }

    // Quick check to determine if target is callable, in the spec
    // this throws a TypeError, but we will just return undefined.
    if ( !jQueryisFunction( fn ) ) {
        return undefined;
    }

    // Simulated bind
    args = core_slice.call( arguments, 2 );
    proxy = function() {

```

```

        return fn.apply( context || this, args.concat( core_slice.call( arguments ) )
) );
};

// Set the guid of unique handler to the same of original handler, so it can
be removed
proxy.guid = fn.guid = fn.guid || jQuery.guid++;

return proxy;
},

// Multifunctional method to get and set values of a collection
// The value/s can optionally be executed if it's a function
access: function( elems, fn, key, value, chainable, emptyGet, raw ) {
    var i = 0,
        length = elems.length,
        bulk = key == null;

    // Sets many values
    if ( jQuery.type( key ) === "object" ) {
        chainable = true;
        for ( i in key ) {
            jQuery.access( elems, fn, i, key[i], true, emptyGet, raw );
        }
    }

    // Sets one value
} else if ( value !== undefined ) {
    chainable = true;

    if ( !jQueryisFunction( value ) ) {
        raw = true;
    }

    if ( bulk ) {
        // Bulk operations run against the entire set
        if ( raw ) {
            fn.call( elems, value );
            fn = null;

            // ...except when executing function values
        } else {
            bulk = fn;
            fn = function( elem, key, value ) {
                return bulk.call( jQuery( elem ), value );
            };
        }
    }
}

```

```

        }
    }

    if ( fn ) {
        for ( ; i < length; i++ ) {
            fn( elems[i], key, raw ? value : value.call( elems[i], i,
fn( elems[i], key ) ) );
        }
    }
}

return chainable ?
    elems :

    // Gets
    bulk ?
        fn.call( elems ) :
        length ? fn( elems[0], key ) : emptyGet;
},
now: function() {
    return ( new Date() ).getTime();
},

// A method for quickly swapping in/out CSS properties to get correct calculations.
// Note: this method belongs to the css module but it's needed here for the
support module.
// If support gets modularized, this method should be moved back to the css
module.
swap: function( elem, options, callback, args ) {
    var ret, name,
        old = {};

    // Remember the old values, and insert the new ones
    for ( name in options ) {
        old[ name ] = elem.style[ name ];
        elem.style[ name ] = options[ name ];
    }

    ret = callback.apply( elem, args || [] );

    // Revert the old values
    for ( name in options ) {
        elem.style[ name ] = old[ name ];
    }
}

```

```

        return ret;
    }
});

jQuery.ready.promise = function( obj ) {
    if ( !readyList ) {

        readyList = jQuery.Deferred();

        // Catch cases where $(document).ready() is called after the browser event
        // has already occurred.
        // we once tried to use readyState "interactive" here, but it caused issues
        // like the one
        // discovered by ChrisS here: http://bugs.jquery.com/ticket/
        12282#comment:15
        if ( document.readyState === "complete" ) {
            // Handle it asynchronously to allow scripts the opportunity to delay
            ready
            setTimeout( jQuery.ready );

            // Standards-based browsers support DOMContentLoaded
        } else if ( document.addEventListener ) {
            // Use the handy event callback
            document.addEventListener( "DOMContentLoaded", completed,
false );
            // A fallback to window.onload, that will always work
            window.addEventListener( "load", completed, false );
        }

        // If IE event model is used
    } else {
        // Ensure firing before onload, maybe late but safe also for iframes
        document.attachEvent( "onreadystatechange", completed );

        // A fallback to window.onload, that will always work
        window.attachEvent( "onload", completed );

        // If IE and not a frame
        // continually check to see if the document is ready
        var top = false;

        try {
            top = window.frameElement == null &&
document.documentElement;

```

```

} catch(e) {}

if ( top && top.doScroll ) {
    (function doScrollCheck() {
        if ( !jQuery.isReady ) {

            try {
                // Use the trick by Diego Perini
                // http://javascript.nwbox.com/
IEContentLoaded/
                top.doScroll("left");
            } catch(e) {
                return setTimeout( doScrollCheck, 50 );
            }

            // detach all dom ready events
            detach();

            // and execute any waiting functions
            jQuery.ready();
        }
    })();
}
}

return readyList.promise( obj );
};

// Populate the class2type map
jQuery.each("Boolean Number String Function Array Date RegExp Object Error".split(" "),
function(i, name) {
    class2type[ "[object " + name + "]" ] = name.toLowerCase();
});

function isArraylike( obj ) {
    var length = obj.length,
        type = jQuery.type( obj );

    if ( jQuery.isWindow( obj ) ) {
        return false;
    }

    if ( obj.nodeType === 1 && length ) {
        return true;
    }
}

```

```
return type === "array" || type !== "function" &&
( length === 0 ||
typeof length === "number" && length > 0 && ( length - 1 ) in obj );
}

// All jQuery objects should point back to these
rootjQuery = jQuery(document);
/*!
 * Sizzle CSS Selector Engine v1.10.2
 * http://sizzlejs.com/
 *
 * Copyright 2013 jQuery Foundation, Inc. and other contributors
 * Released under the MIT license
 * http://jquery.org/license
 *
 * Date: 2013-07-03
 */
(function( window, undefined ) {

var i,
    support,
    cachedruns,
    Expr,
    getText,
    isXML,
    compile,
    outermostContext,
    sortInput,

    // Local document vars
    setDocument,
    document,
    docElem,
    documentIsHTML,
    rbuggyQSA,
    rbuggyMatches,
    matches,
    contains,

    // Instance-specific data
    expando = "sizzle" + -(new Date()),
    preferredDoc = window.document,
    dirruns = 0,
    done = 0,
```

```

classCache = createCache(),
tokenCache = createCache(),
compilerCache = createCache(),
hasDuplicate = false,
sortOrder = function( a, b ) {
    if ( a === b ) {
        hasDuplicate = true;
        return 0;
    }
    return 0;
},

// General-purpose constants
strundefined = typeof undefined,
MAX_NEGATIVE = 1 << 31,

// Instance methods
hasOwn = {}.hasOwnProperty,
arr = [],
pop = arr.pop,
push_native = arr.push,
push = arr.push,
slice = arr.slice,
// Use a stripped-down indexOf if we can't use a native one
indexOf = arr.indexOf || function( elem ) {
    var i = 0,
        len = this.length;
    for ( ; i < len; i++ ) {
        if ( this[i] === elem ) {
            return i;
        }
    }
    return -1;
},
booleans = "checked|selected|async|autofocus|autoplay|controls|defer|disabled|hidden|ismap|loop|multiple|open|readonly|required|scoped",
// Regular expressions
// Whitespace characters http://www.w3.org/TR/css3-selectors/#whitespace
whitespace = "[\\x20\\t\\r\\n\\f]",
// http://www.w3.org/TR/css3-syntax/#characters
characterEncoding = "(?:\\\\.|[\\w-]|[^-\\x00-\\xa0])+",

```

```

// Loosely modeled on CSS identifier characters
// An unquoted value should be a CSS identifier http://www.w3.org/TR/css3-
selectors/#attribute-selectors
// Proper syntax: http://www.w3.org/TR/CSS21/syndata.html#value-def-identifier
identifier = characterEncoding.replace( "w", "w#" ),

// Acceptable operators http://www.w3.org/TR/selectors/#attribute-selectors
attributes = "\\[" + whitespace + "(" + characterEncoding + ")" + whitespace +
  "*(:([*^$!~]?=)" + whitespace + "*(:(['\"])((?:\\\\.|[^\\"\\])*)?)\\3|(" + identifier
+ ")\\)" + whitespace + "*\\]",

// Prefer arguments quoted,
// then not containing pseudos/brackets,
// then attribute selectors/non-parenthetical expressions,
// then anything else
// These preferences are here to reduce the number of selectors
// needing tokenize in the PSEUDO preFilter
pseudos = ":" + characterEncoding + ")(:\\(((['\"])((?:\\\\.|[^\\"\\])*)?)\\3|((?:\\\\.|[^\\"\\
\\0[\\"]])" + attributes.replace( 3, 8 ) + ")*\\.)\\)",

// Leading and non-escaped trailing whitespace, capturing some non-whitespace
// characters preceding the latter
rtrim = new RegExp( "^" + whitespace + "+|((?:^|[^\\"\\])(?:\\\\.)*)" + whitespace +
  "+$", "g" ),

rcomma = new RegExp( "^" + whitespace + "*", + whitespace + "*" ),
rcombinators = new RegExp( "^" + whitespace + "*([>+~]|" + whitespace + ")" +
  whitespace + "*" ),

rsibling = new RegExp( whitespace + "*[+~]" ),
rattributeQuotes = new RegExp( "=" + whitespace + "*([^\\"\\]\"\")*" + whitespace +
  "*\\]", "g" ),

rpseudo = new RegExp( pseudos ),
ridentifier = new RegExp( "^" + identifier + "$" ),

matchExpr = {
  "ID": new RegExp( "^#(" + characterEncoding + ")" ),
  "CLASS": new RegExp( "^\\.(" + characterEncoding + ")" ),
  "TAG": new RegExp( "^(" + characterEncoding.replace( "w", "w*" ) + ")" ),
  "ATTR": new RegExp( "^" + attributes ),
  "PSEUDO": new RegExp( "^" + pseudos ),
  "CHILD": new RegExp( "^:(onlyfirstlast\\nth\\nth-last)-(childof-type)(?:\\(" +
  whitespace +

```

```

    +"*(evenlodd|([+-]|)(\\d*)n|)" + whitespace + "*(?:([+-]|)" + whitespace
+
    +"*(\\d+|))" + whitespace + "*\\(|", "i" ),
"bool": new RegExp( "^(?::" + booleans + ")$", "i" ),
// For use in libraries implementing .is()
// We use this for POS matching in `select`
"needsContext": new RegExp( "^" + whitespace + "*[>+~]|:(evenlodd|ql
gt|lt|nth|first|last)(?:\\(" +
    whitespace + "*(?-\\d)?\\d*)" + whitespace + "*\\(|)?=[^-]|\$)", "i" )
},
rnative = /^[^{}]+\\s*[native \\w/,
// Easily-parseable/retrievable ID or TAG or CLASS selectors
rquickExpr = /^(?:#([\\w-]+)|(\\w+)|\\.([\\w-]+))$/,
rinputs = /^(:input|select|text|area|button)$|/i,
rheader = /^h\\d$/i,
rescape = '/\\\\g',
// CSS escapes http://www.w3.org/TR/CSS21/syndata.html#escaped-characters
runescape = new RegExp( "\\\\da-f]{1,6}" + whitespace + "?|(" + whitespace +
")|.", "ig" ),
funescape = function( _, escaped, escapedWhitespace ) {
    var high = "0x" + escaped - 0x10000;
    // NaN means non-codepoint
    // Support: Firefox
    // Workaround erroneous numeric interpretation of +"0x"
    return high !== high || escapedWhitespace ?
        escaped :
        // BMP codepoint
        high < 0 ?
            String.fromCharCode( high + 0x10000 ) :
            // Supplemental Plane codepoint (surrogate pair)
            String.fromCharCode( high >> 10 | 0xD800, high & 0x3FF | 0xDC00 );
    };
// Optimize for push.apply( _, NodeList )
try {
    push.apply(
        (arr = slice.call( preferredDoc.childNodes )),
        preferredDoc.childNodes
    );
}

```

```

// Support: Android<4.0
// Detect silently failing push.apply
arr[ preferredDoc.childNodes.length ].nodeType;
} catch ( e ) {
    push = { apply: arr.length ?

        // Leverage slice if possible
        function( target, els ) {
            push_native.apply( target, slice.call(els) );
        } :

        // Support: IE<9
        // Otherwise append directly
        function( target, els ) {
            var j = target.length,
                i = 0;
            // Can't trust NodeList.length
            while ( (target[j++] = els[i++]) ) {}
            target.length = j - 1;
        }
    };
}

function Sizzle( selector, context, results, seed ) {
    var match, elem, m,.nodeType,
        // QSA vars
        i, groups, old, nid, newContext, newSelector;

    if ( ( context ? context.ownerDocument || context : preferredDoc ) !== document ) {
        setDocument( context );
    }

    context = context || document;
    results = results || [];

    if ( !selector || typeof selector !== "string" ) {
        return results;
    }

    if ( (nodeType = context.nodeType) !== 1 && nodeType !== 9 ) {
        return [];
    }

    if ( documentIsHTML && !seed ) {

```

```

// Shortcuts
if ( (match = rquickExpr.exec( selector )) ) {
    // Speed-up: Sizzle("#ID")
    if ( (m = match[1]) ) {
        if ( nodeType === 9 ) {
            elem = context.getElementById( m );
            // Check parentNode to catch when Blackberry 4.6
            returns
                // nodes that are no longer in the document #6963
                if ( elem && elem.parentNode ) {
                    // Handle the case where IE, Opera, and Webkit
            return items
                // by name instead of ID
                if ( elem.id === m ) {
                    results.push( elem );
                    return results;
                }
            } else {
                return results;
            }
        } else {
            // Context is not a document
            if ( context.ownerDocument && (elem =
context.ownerDocument.getElementById( m )) &&
                contains( context, elem ) && elem.id === m ) {
                    results.push( elem );
                    return results;
                }
            }
        }
    }

    // Speed-up: Sizzle("TAG")
} else if ( match[2] ) {
    push.apply( results, context.getElementsByTagName( selector )
);
    return results;

    // Speed-up: Sizzle(".CLASS")
} else if ( (m = match[3]) && support.getElementsByClassName &&
context.getElementsByClassName ) {
    push.apply( results, context.getElementsByClassName( m ) );
    return results;
}
}

// QSA path

```

```

if ( support.qsa && (!rbuggyQSA || !rbuggyQSA.test( selector )) ) {
    nid = old = expando;
    newContext = context;
    newSelector = nodeType === 9 && selector;

    // qSA works strangely on Element-rooted queries
    // We can work around this by specifying an extra ID on the root
    // and working up from there (Thanks to Andrew Dupont for the
technique)
    // IE 8 doesn't work on object elements
    if ( nodeType === 1 && context.nodeName.toLowerCase() !==
"object" ) {
        groups = tokenize( selector );

        if ( (old = context.getAttribute("id")) ) {
            nid = old.replace( rescape, "\\$&" );
        } else {
            context.setAttribute( "id", nid );
        }
        nid = "[id='" + nid + "'] ";

        i = groups.length;
        while ( i-- ) {
            groups[i] = nid + toSelector( groups[i] );
        }
        newContext = rsibling.test( selector ) && context.parentNode ||
context;
        newSelector = groups.join(",");
    }

    if ( newSelector ) {
        try {
            push.apply( results,
                newContext.querySelectorAll( newSelector )
            );
            return results;
        } catch(qsaError) {
        } finally {
            if ( !old ) {
                context.removeAttribute("id");
            }
        }
    }
}

```

```

// All others
return select( selector.replace( rtrim, "$1" ), context, results, seed );
}

/**
 * Create key-value caches of limited size
 * @returns {Function(string, Object)} Returns the Object data after storing it on itself with
 *         property name the (space-suffixed) string and (if the cache is larger than
Expr.cacheLength)
 *         deleting the oldest entry
 */
function createCache() {
    var keys = [];

    function cache( key, value ) {
        // Use (key + " ") to avoid collision with native prototype properties (see
Issue #157)
        if ( keys.push( key += " " ) > Expr.cacheLength ) {
            // Only keep the most recent entries
            delete cache[ keys.shift() ];
        }
        return (cache[ key ] = value);
    }
    return cache;
}

/**
 * Mark a function for special use by Sizzle
 * @param {Function} fn The function to mark
 */
function markFunction( fn ) {
    fn[ expando ] = true;
    return fn;
}

/**
 * Support testing using an element
 * @param {Function} fn Passed the created div and expects a boolean result
 */
function assert( fn ) {
    var div = document.createElement("div");

    try {
        return !!fn( div );
    }
}

```

```

} catch (e) {
    return false;
} finally {
    // Remove from its parent by default
    if ( div.parentNode ) {
        div.parentNode.removeChild( div );
    }
    // release memory in IE
    div = null;
}
}

/***
 * Adds the same handler for all of the specified attrs
 * @param {String} attrs Pipe-separated list of attributes
 * @param {Function} handler The method that will be applied
 */
function addHandle( attrs, handler ) {
    var arr = attrs.split("|"),
        i = attrs.length;

    while ( i-- ) {
        Expr.attrHandle[ arr[i] ] = handler;
    }
}

/***
 * Checks document order of two siblings
 * @param {Element} a
 * @param {Element} b
 * @returns {Number} Returns less than 0 if a precedes b, greater than 0 if a follows b
 */
function siblingCheck( a, b ) {
    var cur = b && a,
        diff = cur && a.nodeType === 1 && b.nodeType === 1 &&
            ( ~b.sourceIndex || MAX_NEGATIVE ) -
            ( ~a.sourceIndex || MAX_NEGATIVE );

    // Use IE sourceIndex if available on both nodes
    if ( diff ) {
        return diff;
    }

    // Check if b follows a
    if ( cur ) {

```

```

        while ( (cur = cur.nextSibling) ) {
            if ( cur === b ) {
                return -1;
            }
        }

        return a ? 1 : -1;
    }

/***
 * Returns a function to use in pseudos for input types
 * @param {String} type
 */
function createInputPseudo( type ) {
    return function( elem ) {
        var name = elem.nodeName.toLowerCase();
        return name === "input" && elem.type === type;
    };
}

/***
 * Returns a function to use in pseudos for buttons
 * @param {String} type
 */
function createButtonPseudo( type ) {
    return function( elem ) {
        var name = elem.nodeName.toLowerCase();
        return (name === "input" || name === "button") && elem.type === type;
    };
}

/***
 * Returns a function to use in pseudos for positionals
 * @param {Function} fn
 */
function createPositionalPseudo( fn ) {
    return markFunction(function( argument ) {
        argument = +argument;
        return markFunction(function( seed, matches ) {
            var j,
                matchIndexes = fn( [], seed.length, argument ),
                i = matchIndexes.length;

            // Match elements found at the specified indexes
        });
    });
}

```

```

        while ( i-- ) {
            if ( seed[ ( j = matchIndexes[i] ) ] ) {
                seed[j] = !(matches[j] = seed[j]);
            }
        }
    });

});

/***
 * Detect xml
 * @param {Element|Object} elem An element or a document
 */
isXML = Sizzle.isXML = function( elem ) {
    // documentElement is verified for cases where it doesn't yet exist
    // (such as loading iframes in IE - #4833)
    var documentElement = elem && (elem.ownerDocument ||
elem).documentElement;
    return documentElement ? documentElement.nodeName !== "HTML" : false;
};

// Expose support vars for convenience
support = Sizzle.support = {};

/***
 * Sets document-related variables once based on the current document
 * @param {Element|Object} [doc] An element or document object to use to set the
document
 * @returns {Object} Returns the current document
 */
setDocument = Sizzle.setDocument = function( node ) {
    var doc = node ? node.ownerDocument || node : preferredDoc,
        parent = doc.defaultView;

    // If no document and documentElement is available, return
    if ( doc === document || doc.nodeType !== 9 || !doc.documentElement ) {
        return document;
    }

    // Set our document
    document = doc;
    docElem = doc.documentElement;

    // Support tests
    documentIsHTML = !isXML( doc );
}

```

```

// Support: IE>8
// If iframe document is assigned to "document" variable and if iframe has been
reloaded,
    // IE will throw "permission denied" error when accessing "document" variable,
see jQuery #13936
    // IE6-8 do not support the defaultView property so parent will be undefined
if ( parent && parent.attachEvent && parent !== parent.top ) {
    parent.attachEvent( "onbeforeunload", function() {
        setDocument();
    });
}

/* Attributes
----- */

// Support: IE<8
// Verify that getAttribute really returns attributes and not properties (excepting IE8
booleans)
support.attributes = assert(function( div ) {
    div.className = "i";
    return !div.getAttribute("className");
});

/* getElement(s)By*
----- */

// Check if getElementsByTagName("*") returns only elements
support.getElementsByTagName = assert(function( div ) {
    div.appendChild( doc.createComment(" ") );
    return !div.getElementsByTagName("*").length;
});

// Check if getElementsByClassName can be trusted
support.getElementsByClassName = assert(function( div ) {
    div.innerHTML = "<div class='a'></div><div class='a i'></div>";
    // Support: Safari<4
    // Catch class over-caching
    div.firstChild.className = "i";
    // Support: Opera<10
    // Catch gECBN failure to find non-leading classes
    return div.getElementsByClassName("i").length === 2;
});

```

```

// Support: IE<10
// Check if getElementById returns elements by name
// The broken getElementById methods don't pick up programatically-set names,
// so use a roundabout getElementsByName test
support.getById = assert(function( div ) {
    docElem.appendChild( div ).id = expando;
    return !doc.getElementsByName || !
doc.getElementsByName( expando ).length;
});

// ID find and filter
if ( support.getById ) {
    Expr.find["ID"] = function( id, context ) {
        if ( typeof context.getElementById !== strundefined &&
documentIsHTML ) {
            var m = context.getElementById( id );
            // Check parentNode to catch when Blackberry 4.6 returns
            // nodes that are no longer in the document #6963
            return m && m.parentNode ? [m] : [];
        }
    };
    Expr.filter["ID"] = function( id ) {
        var attrId = id.replace( runescape, funescape );
        return function( elem ) {
            return elem.getAttribute("id") === attrId;
        };
    };
} else {
    // Support: IE6/7
    // getElementById is not reliable as a find shortcut
    delete Expr.find["ID"];

    Expr.filter["ID"] = function( id ) {
        var attrId = id.replace( runescape, funescape );
        return function( elem ) {
            var node = typeof elem.getAttributeNode !== strundefined
&& elem.getAttributeNode("id");
            return node && node.value === attrId;
        };
    };
}

// Tag
Expr.find["TAG"] = support.getElementsByTagName ?
    function( tag, context ) {

```

```

if ( typeof context.getElementsByTagName !== strundefined ) {
    return context.getElementsByTagName( tag );
}
}:
function( tag, context ) {
    var elem,
        tmp = [],
        i = 0,
        results = context.getElementsByTagName( tag );

    // Filter out possible comments
    if ( tag === "*" ) {
        while ( (elem = results[i++]) ) {
            if ( elem.nodeType === 1 ) {
                tmp.push( elem );
            }
        }
        return tmp;
    }
    return results;
};

// Class
Expr.find["CLASS"] = support.getElementsByClassName && function( className,
context ) {
    if ( typeof context.getElementsByClassName !== strundefined &&
documentIsHTML ) {
        return context.getElementsByClassName( className );
    }
};

/* QSA/matchesSelector
----- */
// QSA and matchesSelector support

// matchesSelector(:active) reports false when true (IE9/Opera 11.5)
rbuggyMatches = [];

// qSa(:focus) reports false when true (Chrome 21)
// We allow this because of a bug in IE8/9 that throws an error
// whenever `document.activeElement` is accessed on an iframe
// So, we allow :focus to pass through QSA all the time to avoid the IE error
// See http://bugs.jquery.com/ticket/13378

```

```

rbuggyQSA = [];

if ( (support.qsa = rnative.test( doc.querySelectorAll )) ) {
    // Build QSA regex
    // Regex strategy adopted from Diego Perini
    assert(function( div ) {
        // Select is set to empty string on purpose
        // This is to test IE's treatment of not explicitly
        // setting a boolean content attribute,
        // since its presence should be enough
        // http://bugs.jquery.com/ticket/12359
        div.innerHTML = "<select><option selected='></option></
select>";

        // Support: IE8
        // Boolean attributes and "value" are not treated correctly
        if ( !div.querySelectorAll("[selected]").length ) {
            rbuggyQSA.push( "\\[" + whitespace + "*(?:value|"
+ ")" );
        }
    });

    // Webkit/Opera - :checked should return selected option elements
    // http://www.w3.org/TR/2011/REC-css3-selectors-20110929/
#checked

    // IE8 throws error here and will not see later tests
    if ( !div.querySelectorAll(":checked").length ) {
        rbuggyQSA.push(":checked");
    }
});

assert(function( div ) {

    // Support: Opera 10-12/IE8
    // ^= $= *= and empty values
    // Should not select anything
    // Support: Windows 8 Native Apps
    // The type attribute is restricted during .innerHTML assignment
    var input = doc.createElement("input");
    input.setAttribute( "type", "hidden" );
    div.appendChild( input ).setAttribute( "t", "" );

    if ( div.querySelectorAll("[t^='']").length ) {
        rbuggyQSA.push( "[*^$]= " + whitespace + "*(?:\"\\\")" );
    }
});

```

```

// FF 3.5 - :enabled/:disabled and hidden elements (hidden elements
are still enabled)
    // IE8 throws error here and will not see later tests
    if ( !div.querySelectorAll(":enabled").length ) {
        rbuggyQSA.push( ":enabled", ":disabled" );
    }

    // Opera 10-11 does not throw on post-comma invalid pseudos
    div.querySelectorAll("*,:x");
    rbuggyQSA.push(", *:");

};

}

if ( (support.matchesSelector = rnative.test( (matches =
docElem.webkitMatchesSelector ||
docElem.mozMatchesSelector ||
docElem.oMatchesSelector ||
docElem.msMatchesSelector) )) ) {

    assert(function( div ) {
        // Check to see if it's possible to do matchesSelector
        // on a disconnected node (IE 9)
        support.disconnectedMatch = matches.call( div, "div" );

        // This should fail with an exception
        // Gecko does not error, returns false instead
        matches.call( div, "[s!='']:x" );
        rbuggyMatches.push( "!=" , pseudos );
    });
}

rbuggyQSA = rbuggyQSA.length && new RegExp( rbuggyQSA.join("|") );
rbuggyMatches = rbuggyMatches.length && new RegExp( rbuggyMatches.join("|") );
};

/* Contains
----- */
// Element contains another
// Purposefully does not implement inclusive descendent
// As in, an element does not contain itself
contains = rnative.test( docElem.contains ) || docElem.compareDocumentPosition ?
    function( a, b ) {
        var adown = a.nodeType === 9 ? a.documentElement : a,
            bup = b && b.parentNode;

```

```

        return a === bup || !( bup && bup.nodeType === 1 && (
            adown.contains ?
                adown.contains( bup ) :
                a.compareDocumentPosition &&
            a.compareDocumentPosition( bup ) & 16
        )));
    }:
    function( a, b ){
        if( b ){
            while ( (b = b.parentNode) ){
                if( b === a ){
                    return true;
                }
            }
        }
        return false;
    };
/* Sorting
----- */

// Document order sorting
sortOrder = docElem.compareDocumentPosition ?
function( a, b ){

    // Flag for duplicate removal
    if( a === b ){
        hasDuplicate = true;
        return 0;
    }

    var compare = b.compareDocumentPosition &&
a.compareDocumentPosition && a.compareDocumentPosition( b );

    if( compare ){
        // Disconnected nodes
        if( compare & 1 ||
            (!support.sortDetached && b.compareDocumentPosition( a )
== compare) ){

            // Choose the first element that is related to our preferred
document
            if( a === doc || contains(preferredDoc, a) ){
                return -1;
            }
        }
    }
}

```

```

        if ( b === doc || contains(preferredDoc, b) ) {
            return 1;
        }

        // Maintain original order
        return sortInput ?
            ( indexOf.call( sortInput, a ) - indexOf.call( sortInput, b ) )
        :
            0;
    }

    return compare & 4 ? -1 : 1;
}

// Not directly comparable, sort on existence of method
return a.compareDocumentPosition ? -1 : 1;
}:
function( a, b ) {
    var cur,
        i = 0,
        aup = a.parentNode,
        bup = b.parentNode,
        ap = [ a ],
        bp = [ b ];

    // Exit early if the nodes are identical
    if ( a === b ) {
        hasDuplicate = true;
        return 0;

    // Parentless nodes are either documents or disconnected
    } else if ( !aup || !bup ) {
        return a === doc ? -1 :
            b === doc ? 1 :
            aup ? -1 :
            bup ? 1 :
            sortInput ?
                ( indexOf.call( sortInput, a ) - indexOf.call( sortInput, b ) ) :
            0;

    // If the nodes are siblings, we can do a quick check
    } else if ( aup === bup ) {
        return siblingCheck( a, b );
    }
}

```

```

// Otherwise we need full lists of their ancestors for comparison
cur = a;
while ( (cur = cur.parentNode) ) {
    ap.unshift( cur );
}
cur = b;
while ( (cur = cur.parentNode) ) {
    bp.unshift( cur );
}

// Walk down the tree looking for a discrepancy
while ( ap[i] === bp[i] ) {
    i++;
}

return i ?
    // Do a sibling check if the nodes have a common ancestor
    siblingCheck( ap[i], bp[i] ) :

    // Otherwise nodes in our document sort first
    ap[i] === preferredDoc ? -1 :
    bp[i] === preferredDoc ? 1 :
    0;
};

return doc;
};

Sizzle.matches = function( expr, elements ) {
    return Sizzle( expr, null, null, elements );
};

Sizzle.matchesSelector = function( elem, expr ) {
    // Set document vars if needed
    if ( ( elem.ownerDocument || elem ) !== document ) {
        setDocument( elem );
    }

    // Make sure that attribute selectors are quoted
    expr = expr.replace( rattributeQuotes, "='$1'" );

    if ( support.matchesSelector && documentIsHTML &&
        ( !rbuggyMatches || !rbuggyMatches.test( expr ) ) &&
        ( !rbuggyQSA || !rbuggyQSA.test( expr ) ) ) {

```

```

try {
    var ret = matches.call( elem, expr );

    // IE 9's matchesSelector returns false on disconnected nodes
    if ( ret || support.disconnectedMatch ||
         // As well, disconnected nodes are said to be in a
document
        // fragment in IE 9
        elem.document && elem.document.nodeType !== 11 ) {
        return ret;
    }
} catch(e) {}

return Sizzle( expr, document, null, [elem] ).length > 0;
};

Sizzle.contains = function( context, elem ) {
    // Set document vars if needed
    if ( ( context.ownerDocument || context ) !== document ) {
        setDocument( context );
    }
    return contains( context, elem );
};

Sizzle.attr = function( elem, name ) {
    // Set document vars if needed
    if ( ( elem.ownerDocument || elem ) !== document ) {
        setDocument( elem );
    }

var fn = Expr.attrHandle[ name.toLowerCase() ],
    // Don't get fooled by Object.prototype properties (jQuery #13807)
    val = fn && hasOwn.call( Expr.attrHandle, name.toLowerCase() ) ?
        fn( elem, name, !documentIsHTML ) :
        undefined;

    return val === undefined ?
        support.attributes || !documentIsHTML ?
            elem.getAttribute( name ) :
            (val = elem.getAttributeNode(name)) && val.specified ?
                val.value :
                null :
                val;
};

```

```

Sizzle.error = function( msg ) {
    throw new Error( "Syntax error, unrecognized expression: " + msg );
};

/***
 * Document sorting and removing duplicates
 * @param {ArrayLike} results
 */
Sizzle.uniqueSort = function( results ) {
    var elem,
        duplicates = [],
        j = 0,
        i = 0;

    // Unless we *know* we can detect duplicates, assume their presence
    hasDuplicate = !support.detectDuplicates;
    sortInput = !support.sortStable && results.slice( 0 );
    results.sort( sortOrder );

    if ( hasDuplicate ) {
        while ( (elem = results[i++]) ) {
            if ( elem === results[ i ] ) {
                j = duplicates.push( i );
            }
        }
        while ( j-- ) {
            results.splice( duplicates[ j ], 1 );
        }
    }

    return results;
};

/***
 * Utility function for retrieving the text value of an array of DOM nodes
 * @param {ArrayElement} elem
 */
getText = Sizzle.getText = function( elem ) {
    var node,
        ret = "",
        i = 0,
        nodeType = elem.nodeType;

    if ( !nodeType ) {

```

```

// If no nodeType, this is expected to be an array
for ( ; (node = elem[i]); i++ ) {
    // Do not traverse comment nodes
    ret += getText( node );
}
} else if ( nodeType === 1 || nodeType === 9 || nodeType === 11 ) {
    // Use textContent for elements
    // innerText usage removed for consistency of new lines (see #11153)
    if ( typeof elem.textContent === "string" ) {
        return elem.textContent;
    } else {
        // Traverse its children
        for ( elem = elem.firstChild; elem; elem = elem.nextSibling ) {
            ret += getText( elem );
        }
    }
} else if ( nodeType === 3 || nodeType === 4 ) {
    return elem.nodeValue;
}
// Do not include comment or processing instruction nodes

return ret;
};

```

```

Expr = Sizzle.selectors = {

    // Can be adjusted by the user
    cacheLength: 50,

    createPseudo: markFunction,

    match: matchExpr,

    attrHandle: {},

    find: {},

    relative: {
        ">": { dir: "parentNode", first: true },
        " ": { dir: "parentNode" },
        "+": { dir: "previousSibling", first: true },
        "~": { dir: "previousSibling" }
    },
    preFilter: {}
}
```

```

"ATTR": function( match ) {
    match[1] = match[1].replace( runescape, funescape );

    // Move the given value to match[3] whether quoted or unquoted
    match[3] = ( match[4] || match[5] || "" ).replace( runescape,
funescape );

    if ( match[2] === "~=" ) {
        match[3] = " " + match[3] + " ";
    }

    return match.slice( 0, 4 );
},

"CHILD": function( match ) {
    /* matches from matchExpr["CHILD"]
     * 1 type (onlyInthl...)
     * 2 what (childof-type)
     * 3 argument (evenlodd\|d*\|d*n([+-]\|d+)?|...)
     * 4 xn-component of xn+y argument ([+-]?d*n|)
     * 5 sign of xn-component
     * 6 x of xn-component
     * 7 sign of y-component
     * 8 y of y-component
    */
    match[1] = match[1].toLowerCase();

    if ( match[1].slice( 0, 3 ) === "nth" ) {
        // nth-* requires argument
        if ( !match[3] ) {
            Sizzle.error( match[0] );
        }

        // numeric x and y parameters for Expr.filter.CHILD
        // remember that false/true cast respectively to 0/1
        match[4] = +( match[4] ? match[5] + (match[6] || 1) : 2 *
( match[3] === "even" || match[3] === "odd" ) );
        match[5] = +( ( match[7] + match[8] ) || match[3] === "odd" );

        // other types prohibit arguments
    } else if ( match[3] ) {
        Sizzle.error( match[0] );
    }

    return match;
}

```

```

    },
    "PSEUDO": function( match ) {
        var excess,
            unquoted = !match[5] && match[2];

        if ( matchExpr["CHILD"].test( match[0] ) ) {
            return null;
        }

        // Accept quoted arguments as-is
        if ( match[3] && match[4] !== undefined ) {
            match[2] = match[4];
        }

        // Strip excess characters from unquoted arguments
    } else if ( unquoted && rpseudo.test( unquoted ) &&
        // Get excess from tokenize (recursively)
        (excess = tokenize( unquoted, true )) &&
        // advance to the next closing parenthesis
        (excess = unquoted.indexOf( ")" , unquoted.length - excess ) -
unquoted.length ) {

        // excess is a negative index
        match[0] = match[0].slice( 0, excess );
        match[2] = unquoted.slice( 0, excess );
    }

    // Return only captures needed by the pseudo filter method (type
and argument)
    return match.slice( 0, 3 );
}
},
filter: {

    "TAG": function( nodeNameSelector ) {
        var nodeName = nodeNameSelector.replace( runescape,
unescape ).toLowerCase();
        return nodeNameSelector === "*" ?
            function() { return true; } :
            function( elem ) {
                return elem.nodeName &&
elem.nodeName.toLowerCase() === nodeName;
            };
    },
}

```

```

"CLASS": function( className ) {
    var pattern = classCache[ className + " " ];

    return pattern ||
        (pattern = new RegExp( "(^| " + whitespace + ")" + className
+ "(" + whitespace + "|$)" ) ) &&
            classCache( className, function( elem ) {
                return pattern.test( typeof elem.className ===
"string" && elem.className || typeof elem.getAttribute !== strundefined &&
elem.getAttribute("class") || "" );
            });
    },

"ATTR": function( name, operator, check ) {
    return function( elem ) {
        var result = Sizzle.attr( elem, name );

        if ( result == null ) {
            return operator === "!=";
        }
        if ( !operator ) {
            return true;
        }

        result += " ";

        return operator === "=" ? result === check :
            operator === "!=" ? result !== check :
            operator === "^=" ? check && result.indexOf( check )
==== 0 :
            operator === "*=" ? check && result.indexOf( check ) >
-1 :
            operator === "$=" ? check && result.slice( -
check.length ) === check :
            operator === "~=" ? ( " " + result + "
" ).indexOf( check ) > -1 :
                operator === "|=" ? result === check || result.slice( 0,
check.length + 1 ) === check + "-" :
                    false;
    };
},
"CHILD": function( type, what, argument, first, last ) {
    var simple = type.slice( 0, 3 ) !== "nth",

```

```

forward = type.slice( -4 ) !== "last",
ofType = what === "of-type";

return first === 1 && last === 0 ?

// Shortcut for :nth-*(n)
function( elem ) {
    return !!elem.parentNode;
} :

function( elem, context, xml ) {
    var cache, outerCache, node, diff, nodeIndex, start,
        dir = simple !== forward ? "nextSibling" :
    "previousSibling",
        parent = elem.parentNode,
        name = ofType &&
    elem.nodeName.toLowerCase(),
        useCache = !xml && !ofType;

    if ( parent ) {

        // :(first|last|only)-(child|of-type)
        if ( simple ) {
            while ( dir ) {
                node = elem;
                while ( (node = node[ dir ]) ) {
                    if ( ofType ?
node.nodeName.toLowerCase() === name : node.nodeType === 1 ) {
                        return false;
                    }
                }
            }
            // Reverse direction for :only-* (if we
haven't yet done so)
            start = dir = type === "only" && !
start && "nextSibling";
        }
        return true;
    }

    start = [ forward ? parent.firstChild :
parent.lastChild ];

    // non-xml :nth-child(...) stores cache data on
`parent`
    if ( forward && useCache ) {

```

```

        // Seek `elem` from a previously-cached
index
        outerCache = parent[ expando ] ||
cache = outerCache[ type ] || [];
nodeIndex = cache[0] === dirruns &&
cache[1];
diff = cache[0] === dirruns && cache[2];
node = nodeIndex &&
parent.childNodes[ nodeIndex ];

while ( (node = ++nodeIndex && node &&
node[ dir ] ||

// Fallback to seeking `elem` from
the start
        (diff = nodeIndex = 0) || start.pop() ) )

{
    // When found, cache indexes on
`parent` and break
    if ( node.nodeType === 1 && +
+diff && node === elem ) {
        outerCache[ type ] =
[ dirruns, nodeIndex, diff ];
        break;
    }
}

// Use previously-cached element index if
available
} else if ( useCache && (cache = (elem[ expando ] ||
(elem[ expando ] = {}))[ type ]) && cache[0] === dirruns ) {
    diff = cache[1];

    // xml :nth-child(...) or :nth-last-child(...) or :nth-
last)?-of-type(...)
} else {
    // Use the same loop as above to seek
`elem` from the start
    while ( (node = ++nodeIndex && node &&
node[ dir ] ||

        (diff = nodeIndex = 0) || start.pop() ) )

{

```

```

        if ( ( ofType ?
node.nodeName.toLowerCase() === name : node.nodeType === 1 ) && ++diff ) {
// Cache the index of each
encountered element
        if ( useCache ) {
(node[ expando ] = {})[ type ] = [ dirruns, diff ];
        }

        if ( node === elem ) {
break;
        }
    }
}

// Incorporate the offset, then check against cycle
size
diff -= last;
return diff === first || ( diff % first === 0 && diff /
first >= 0 );
}

},
};

"PSEUDO": function( pseudo, argument ) {
// pseudo-class names are case-insensitive
// http://www.w3.org/TR/selectors/#pseudo-classes
// Prioritize by case sensitivity in case custom pseudos are added with
uppercase letters
// Remember that setFilters inherits from pseudos
var args,
fn = Expr.pseudos[ pseudo ] ||
Expr.setFilters[ pseudo.toLowerCase() ] ||
Sizzle.error( "unsupported pseudo: " + pseudo );

// The user may use createPseudo to indicate that
// arguments are needed to create the filter function
// just as Sizzle does
if ( fn[ expando ] ) {
return fn( argument );
}

// But maintain support for old signatures
if ( fn.length > 1 ) {

```

```

args = [ pseudo, pseudo, "", argument ];
return Expr.setFilters.hasOwnProperty( pseudo.toLowerCase() )
?

markFunction(function( seed, matches ) {
    var idx,
        matched = fn( seed, argument ),
        i = matched.length;
    while ( i-- ) {
        idx = indexOf.call( seed, matched[i] );
        seed[ idx ] = !( matches[ idx ] =
matched[i] );
    }
}) :
function( elem ) {
    return fn( elem, 0, args );
};
}

return fn;
},
};

pseudos: {
// Potentially complex pseudos
"not": markFunction(function( selector ) {
    // Trim the selector passed to compile
    // to avoid treating leading and trailing
    // spaces as combinators
    var input = [],
        results = [],
        matcher = compile( selector.replace( rtrim, "$1" ) );

    return matcher[ expando ] ?
        markFunction(function( seed, matches, context, xml ) {
            var elem,
                unmatched = matcher( seed, null, xml, [] ),
                i = seed.length;

            // Match elements unmatched by `matcher`
            while ( i-- ) {
                if ( (elem = unmatched[i]) ) {
                    seed[i] = !(matches[i] = elem);
                }
            }
        }) :
}
};
```

```

        function( elem, context, xml ) {
            input[0] = elem;
            matcher( input, null, xml, results );
            return !results.pop();
        },
    }),

    "has": markFunction(function( selector ) {
        return function( elem ) {
            return Sizzle( selector, elem ).length > 0;
        };
    }),

    "contains": markFunction(function( text ) {
        return function( elem ) {
            return ( elem.textContent || elem.innerText ||
getText( elem ) ).indexOf( text ) > -1;
        };
    }),

    // "Whether an element is represented by a :lang() selector
    // is based solely on the element's language value
    // being equal to the identifier C,
    // or beginning with the identifier C immediately followed by "-".
    // The matching of C against the element's language value is performed
    case-insensitively.
    // The identifier C does not have to be a valid language name."
    // http://www.w3.org/TR/selectors/#lang-pseudo
    "lang": markFunction( function( lang ) {
        // lang value must be a valid identifier
        if ( !identifier.test(lang || "" ) ) {
            Sizzle.error( "unsupported lang: " + lang );
        }
        lang = lang.replace( runescape, funescape ).toLowerCase();
        return function( elem ) {
            var elemLang;
            do {
                if ( (elemLang = documentIsHTML ?
                    elem.lang :
                    elem.getAttribute("xml:lang") ||
elem.getAttribute("lang")) ) {
                    elemLang = elemLang.toLowerCase();
                    return elemLang === lang ||
elemLang.indexOf( lang + "-" ) === 0;
                }
            }
        };
    });
}

```

```

        }
    } while ( (elem = elem.parentNode) && elem.nodeType ===
1 );
        return false;
    },
}),

// Miscellaneous
"target": function( elem ) {
    var hash = window.location && window.location.hash;
    return hash && hash.slice( 1 ) === elem.id;
},
"root": function( elem ) {
    return elem === docElem;
},
"focus": function( elem ) {
    return elem === document.activeElement && (!document.hasFocus ||
document.hasFocus()) && !(elem.type || elem.href || ~elem.tabIndex);
},
// Boolean properties
"enabled": function( elem ) {
    return elem.disabled === false;
},
"disabled": function( elem ) {
    return elem.disabled === true;
},
"checked": function( elem ) {
    // In CSS3, :checked should return both checked and selected
elements
    // http://www.w3.org/TR/2011/REC-css3-selectors-20110929/
#checked
    var nodeName = elem.nodeName.toLowerCase();
    return (nodeName === "input" && !elem.checked) || (nodeName
=== "option" && !elem.selected);
},
"selected": function( elem ) {
    // Accessing this property makes selected-by-default
    // options in Safari work properly
    if ( elem.parentNode ) {

```

```

        elem.parentNode.selectedIndex;
    }

    return elem.selected === true;
},

// Contents
"empty": function( elem ) {
    // http://www.w3.org/TR/selectors/#empty-pseudo
    // :empty is only affected by element nodes and content
nodes(including text(3), cdata(4)),
    // not comment, processing instructions, or others
    // Thanks to Diego Perini for the nodeName shortcut
    // Greater than "@" means alpha characters (specifically not starting
with "#" or "?")
    for ( elem = elem.firstChild; elem; elem = elem.nextSibling ) {
        if ( elem.nodeName > "@" || elem.nodeType === 3 ||

elem.nodeType === 4 ) {
            return false;
        }
    }
    return true;
},

"parent": function( elem ) {
    return !Expr.pseudos["empty"]( elem );
},

// Element/input types
"header": function( elem ) {
    return rheader.test( elem.nodeName );
},

"input": function( elem ) {
    return rinputs.test( elem.nodeName );
},

"button": function( elem ) {
    var name = elem.nodeName.toLowerCase();
    return name === "input" && elem.type === "button" || name ===
"button";
    },
}

"text": function( elem ) {
    var attr;

```

```

// IE6 and 7 will map elem.type to 'text' for new HTML5 types (search,
etc)
    // use getAttribute instead to test this case
    return elem.nodeName.toLowerCase() === "input" &&
        elem.type === "text" &&
        ( (attr = elem.getAttribute("type")) == null || attr.toLowerCase()
        === elem.type );
    },

// Position-in-collection
"first": createPositionalPseudo(function() {
    return [ 0 ];
}),

"last": createPositionalPseudo(function( matchIndexes, length ) {
    return [ length - 1 ];
}),

"eq": createPositionalPseudo(function( matchIndexes, length, argument ) {
    return [ argument < 0 ? argument + length : argument ];
}),

"even": createPositionalPseudo(function( matchIndexes, length ) {
    var i = 0;
    for ( ; i < length; i += 2 ) {
        matchIndexes.push( i );
    }
    return matchIndexes;
}),

"odd": createPositionalPseudo(function( matchIndexes, length ) {
    var i = 1;
    for ( ; i < length; i += 2 ) {
        matchIndexes.push( i );
    }
    return matchIndexes;
}),

"It": createPositionalPseudo(function( matchIndexes, length, argument ) {
    var i = argument < 0 ? argument + length : argument;
    for ( ; --i >= 0; ) {
        matchIndexes.push( i );
    }
    return matchIndexes;
}),

```

```

"gt": createPositionalPseudo(function( matchIndexes, length, argument ) {
    var i = argument < 0 ? argument + length : argument;
    for ( ; ++i < length; ) {
        matchIndexes.push( i );
    }
    return matchIndexes;
})
};

Expr.pseudos["nth"] = Expr.pseudos["eq"];

// Add button/input type pseudos
for ( i in { radio: true, checkbox: true, file: true, password: true, image: true } ) {
    Expr.pseudos[ i ] = createInputPseudo( i );
}
for ( i in { submit: true, reset: true } ) {
    Expr.pseudos[ i ] = createButtonPseudo( i );
}

// Easy API for creating new setFilters
function setFilters() {}
setFilters.prototype = Expr.filters = Expr.pseudos;
Expr.setFilters = new setFilters();

function tokenize( selector, parseOnly ) {
    var matched, match, tokens, type,
        soFar, groups, preFilters,
        cached = tokenCache[ selector + " " ];

    if ( cached ) {
        return parseOnly ? 0 : cached.slice( 0 );
    }

    soFar = selector;
    groups = [];
    preFilters = Expr.preFilter;

    while ( soFar ) {

        // Comma and first run
        if ( !matched || (match = rcomma.exec( soFar )) ) {
            if ( match ) {
                // Don't consume trailing commas as valid

```

```

        soFar = soFar.slice( match[0].length ) || soFar;
    }
    groups.push( tokens = [] );
}

matched = false;

// Combinators
if ( (match = rcombinators.exec( soFar )) ) {
    matched = match.shift();
    tokens.push({
        value: matched,
        // Cast descendant combinators to space
        type: match[0].replace( rtrim, " " )
    });
    soFar = soFar.slice( matched.length );
}

// Filters
for ( type in Expr.filter ) {
    if ( (match = matchExpr[ type ].exec( soFar )) && (!preFilters[ type ] ||
        (match = preFilters[ type ]( match ))) ) {
        matched = match.shift();
        tokens.push({
            value: matched,
            type: type,
            matches: match
        });
        soFar = soFar.slice( matched.length );
    }
}

if ( !matched ) {
    break;
}
}

// Return the length of the invalid excess
// if we're just parsing
// Otherwise, throw an error or return tokens
return parseOnly ?
    soFar.length :
    soFar ?
        Sizzle.error( selector ) :
        // Cache the tokens

```

```

        tokenCache( selector, groups ).slice( 0 );
    }

function toSelector( tokens ) {
    var i = 0,
        len = tokens.length,
        selector = "";
    for ( ; i < len; i++ ) {
        selector += tokens[i].value;
    }
    return selector;
}

function addCombinator( matcher, combinator, base ) {
    var dir = combinator.dir,
        checkNonElements = base && dir === "parentNode",
        doneName = done++;

    return combinator.first ?
        // Check against closest ancestor/preceding element
        function( elem, context, xml ) {
            while ( (elem = elem[ dir ]) ) {
                if ( elem.nodeType === 1 || checkNonElements ) {
                    return matcher( elem, context, xml );
                }
            }
        } :
        // Check against all ancestor/preceding elements
        function( elem, context, xml ) {
            var data, cache, outerCache,
                dirkey = dirruns + " " + doneName;

            // We can't set arbitrary data on XML nodes, so they don't benefit
from dir caching
            if ( xml ) {
                while ( (elem = elem[ dir ]) ) {
                    if ( elem.nodeType === 1 || checkNonElements ) {
                        if ( matcher( elem, context, xml ) ) {
                            return true;
                        }
                    }
                }
            } else {
                while ( (elem = elem[ dir ]) ) {

```

```

        if ( elem.nodeType === 1 || checkNonElements ) {
            outerCache = elem[ expando ] || (elem[ expando ] =
= {});
            if ( (cache = outerCache[ dir ]) && cache[0] ===
dirkey ) {
                if ( (data = cache[1]) === true || data ===
cachedruns ) {
                    return data === true;
                }
            } else {
                cache = outerCache[ dir ] = [ dirkey ];
                cache[1] = matcher( elem, context, xml ) ||
cachedruns;
                if ( cache[1] === true ) {
                    return true;
                }
            }
        }
    }
};

function elementMatcher( matchers ) {
    return matchers.length > 1 ?
        function( elem, context, xml ) {
            var i = matchers.length;
            while ( i-- ) {
                if ( !matchers[i]( elem, context, xml ) ) {
                    return false;
                }
            }
            return true;
        } :
        matchers[0];
}

function condense( unmatched, map, filter, context, xml ) {
    var elem,
        newUnmatched = [],
        i = 0,
        len = unmatched.length,
        mapped = map != null;

    for ( ; i < len; i++ ) {

```

```

if ( (elem = unmatched[i]) ) {
    if ( !filter || filter( elem, context, xml ) ) {
        newUnmatched.push( elem );
        if ( mapped ) {
            map.push( i );
        }
    }
}
}

return newUnmatched;
}

function setMatcher( preFilter, selector, matcher, postFilter, postFinder, postSelector ) {
    if ( postFilter && !postFilter[ expando ] ) {
        postFilter = setMatcher( postFilter );
    }
    if ( postFinder && !postFinder[ expando ] ) {
        postFinder = setMatcher( postFinder, postSelector );
    }
    return markFunction(function( seed, results, context, xml ) {
        var temp, i, elem,
            preMap = [],
            postMap = [],
            preexisting = results.length,

            // Get initial elements from seed or context
            elems = seed || multipleContexts( selector || "*", context.nodeType ? [
context ] : context, [] ),

            // Prefilter to get matcher input, preserving a map for seed-results
synchronization
        matcherIn = preFilter && ( seed || !selector ) ?
            condense( elems, preMap, preFilter, context, xml ) :
            elems,

        matcherOut = matcher ?
            // If we have a postFinder, or filtered seed, or non-seed
postFilter or preexisting results,
            postFinder || ( seed ? preFilter : preexisting || postFilter ) ?

            // ...intermediate processing is necessary
            [] :

            // ...otherwise use results directly
    
```

```

        results :
        matcherIn;

// Find primary matches
if ( matcher ) {
    matcher( matcherIn, matcherOut, context, xml );
}

// Apply postFilter
if ( postFilter ) {
    temp = condense( matcherOut, postMap );
    postFilter( temp, [], context, xml );

    // Un-match failing elements by moving them back to matcherIn
    i = temp.length;
    while ( i-- ) {
        if ( (elem = temp[i]) ) {
            matcherOut[ postMap[i] ] = !(matcherIn[ postMap[i] ] =
elem);
        }
    }
}

if ( seed ) {
    if ( postFinder || preFilter ) {
        if ( postFinder ) {
            // Get the final matcherOut by condensing this
intermediate into postFinder contexts
            temp = [];
            i = matcherOut.length;
            while ( i-- ) {
                if ( (elem = matcherOut[i]) ) {
                    // Restore matcherIn since elem is not yet
a final match
                    temp.push( (matcherIn[i] = elem) );
                }
            }
            postFinder( null, (matcherOut = []), temp, xml );
        }
    }

    // Move matched elements from seed to results to keep them
synchronized
    i = matcherOut.length;
    while ( i-- ) {
        if ( (elem = matcherOut[i]) &&

```

```

        (temp = postFinder ? indexOf.call( seed, elem ) :
preMap[i]) > -1 ) {

            seed[temp] = !(results[temp] = elem);
        }
    }
}

// Add elements to results, through postFinder if defined
} else {
    matcherOut = condense(
        matcherOut === results ?
            matcherOut.splice( preexisting, matcherOut.length ) :
            matcherOut
    );
    if ( postFinder ) {
        postFinder( null, results, matcherOut, xml );
    } else {
        push.apply( results, matcherOut );
    }
}
});

function matcherFromTokens( tokens ) {
    var checkContext, matcher, j,
        len = tokens.length,
        leadingRelative = Expr.relative[ tokens[0].type ],
        implicitRelative = leadingRelative || Expr.relative[" "],
        i = leadingRelative ? 1 : 0,

        // The foundational matcher ensures that elements are reachable from top-
        // level context(s)
        matchContext = addCombinator( function( elem ) {
            return elem === checkContext;
        }, implicitRelative, true ),
        matchAnyContext = addCombinator( function( elem ) {
            return indexOf.call( checkContext, elem ) > -1;
        }, implicitRelative, true ),
        matchers = [ function( elem, context, xml ) {
            return ( !leadingRelative && ( xml || context !== outermostContext ) ) ||
(
                (checkContext = context).nodeType ?
                    matchContext( elem, context, xml ) :
                    matchAnyContext( elem, context, xml ) );
        }
    ];
}

```

```

    } ];

for ( ; i < len; i++ ) {
    if ( (matcher = Expr.relative[ tokens[i].type ]) ) {
        matchers = [ addCombinator(elementMatcher( matchers ), matcher) ];
    } else {
        matcher = Expr.filter[ tokens[i].type ].apply( null, tokens[i].matches );

        // Return special upon seeing a positional matcher
        if ( matcher[ expando ] ) {
            // Find the next relative operator (if any) for proper handling
            j = ++i;
            for ( ; j < len; j++ ) {
                if ( Expr.relative[ tokens[j].type ] ) {
                    break;
                }
            }
            return setMatcher(
                i > 1 && elementMatcher( matchers ),
                i > 1 && toSelector(
                    // If the preceding token was a descendant
                    combinator, insert an implicit any-element `*`
                    tokens.slice( 0, i - 1 ).concat({ value: tokens[ i - 2 ].type === " " ? "*" : "" })
                ).replace( rtrim, "$1" ),
                matcher,
                i < j && matcherFromTokens( tokens.slice( i, j ) ),
                j < len && matcherFromTokens( (tokens =
tokens.slice( j )) ),
                j < len && toSelector( tokens )
            );
        }
        matchers.push( matcher );
    }
}

return elementMatcher( matchers );
}

function matcherFromGroupMatchers( elementMatchers, setMatchers ) {
    // A counter to specify which element is currently being matched
    var matcherCachedRuns = 0,
        bySet = setMatchers.length > 0,
        byElement = elementMatchers.length > 0,
        superMatcher = function( seed, context, xml, results, expandContext ) {

```

```

var elem, j, matcher,
    setMatched = [],
    matchedCount = 0,
    i = "0",
    unmatched = seed && [],
    outermost = expandContext != null,
    contextBackup = outermostContext,
    // We must always have either seed elements or context
    elems = seed || byElement && Expr.find["TAG"]("*",
expandContext && context.parentNode || context),
    // Use integer dirruns iff this is the outermost matcher
    dirrunsUnique = (dirruns += contextBackup == null ? 1 :
Math.random() || 0.1);

    if ( outermost ) {
        outermostContext = context !== document && context;
        cachedruns = matcherCachedRuns;
    }

    // Add elements passing elementMatchers directly to results
    // Keep `i` a string if there are no elements so `matchedCount` will be
"00" below
    for ( ; (elem = elems[i]) != null; i++ ) {
        if ( byElement && elem ) {
            j = 0;
            while ( (matcher = elementMatchers[j++]) ) {
                if ( matcher( elem, context, xml ) ) {
                    results.push( elem );
                    break;
                }
            }
            if ( outermost ) {
                dirruns = dirrunsUnique;
                cachedruns = ++matcherCachedRuns;
            }
        }

        // Track unmatched elements for set filters
        if ( bySet ) {
            // They will have gone through all possible matchers
            if ( (elem = !matcher && elem) ) {
                matchedCount--;
            }
        }

        // Lengthen the array for every element, matched or not
    }
}

```

```

        if ( seed ) {
            unmatched.push( elem );
        }
    }

// Apply set filters to unmatched elements
matchedCount += i;
if ( bySet && i !== matchedCount ) {
    j = 0;
    while ( (matcher = setMatchers[j++]) ) {
        matcher( unmatched, setMatched, context, xml );
    }

    if ( seed ) {
        // Reintegrate element matches to eliminate the need
for sorting
        if ( matchedCount > 0 ) {
            while ( i-- ) {
                if ( !(unmatched[i] || setMatched[i]) ) {
                    setMatched[i] = pop.call( results );
                }
            }
        }
    }

// Discard index placeholder values to get only actual
matches
setMatched = condense( setMatched );
}

// Add matches to results
push.apply( results, setMatched );

// Seedless set matches succeeding multiple successful
matchers stipulate sorting
if ( outermost && !seed && setMatched.length > 0 &&
    ( matchedCount + setMatchers.length ) > 1 ) {

    Sizzle.uniqueSort( results );
}
}

// Override manipulation of globals by nested matchers
if ( outermost ) {
    dirruns = dirrunsUnique;
}

```

```

        outermostContext = contextBackup;
    }

    return unmatched;
};

return bySet ?
markFunction( superMatcher ) :
superMatcher;
}

compile = Sizzle.compile = function( selector, group /* Internal Use Only */ ) {
    var i,
        setMatchers = [],
        elementMatchers = [],
        cached = compilerCache[ selector + " " ];

    if ( !cached ) {
        // Generate a function of recursive functions that can be used to check each
element
        if ( !group ) {
            group = tokenize( selector );
        }
        i = group.length;
        while ( i-- ) {
            cached = matcherFromTokens( group[i] );
            if ( cached[ expando ] ) {
                setMatchers.push( cached );
            } else {
                elementMatchers.push( cached );
            }
        }
    }

    // Cache the compiled function
    cached = compilerCache( selector,
matcherFromGroupMatchers( elementMatchers, setMatchers ) );
}

return cached;
};

function multipleContexts( selector, contexts, results ) {
    var i = 0,
        len = contexts.length;
    for ( ; i < len; i++ ) {
        Sizzle( selector, contexts[i], results );
    }
}

```

```

}

return results;
}

function select( selector, context, results, seed ) {
    var i, tokens, token, type, find,
        match = tokenize( selector );

    if ( !seed ) {
        // Try to minimize operations if there is only one group
        if ( match.length === 1 ) {

            // Take a shortcut and set the context if the root selector is an ID
            tokens = match[0] = match[0].slice( 0 );
            if ( tokens.length > 2 && (token = tokens[0]).type === "ID" &&
                support.getByld && context.nodeType === 9 &&
                documentIsHTML &&
                Expr.relative[ tokens[1].type ] ) {

                context = ( Expr.find["ID"]
                ( token.matches[0].replace(runescape, funescape), context ) || [] )[0];
                if ( !context ) {
                    return results;
                }
                selector = selector.slice( tokens.shift().value.length );
            }

            // Fetch a seed set for right-to-left matching
            i = matchExpr["needsContext"].test( selector ) ? 0 : tokens.length;
            while ( i-- ) {
                token = tokens[i];

                // Abort if we hit a combinator
                if ( Expr.relative[ (type = token.type) ] ) {
                    break;
                }
                if ( (find = Expr.find[ type ]) ) {
                    // Search, expanding context for leading sibling
combinators
                    if ( (seed = find(
                        token.matches[0].replace( runescape,
funescape ),
                        rsibling.test( tokens[0].type ) &&
context.parentNode || context
                    )) ) {

```

```

        // If seed is empty or no tokens remain, we can
        return early
            tokens.splice( i, 1 );
            selector = seed.length && toSelector( tokens );
            if ( !selector ) {
                push.apply( results, seed );
                return results;
            }

            break;
        }
    }
}

// Compile and execute a filtering function
// Provide `match` to avoid retokenization if we modified the selector above
compile( selector, match ){
    seed,
    context,
    !documentIsHTML,
    results,
    rsibling.test( selector )
};

return results;
}

// One-time assignments

// Sort stability
support.sortStable = expando.split(" ").sort( sortOrder ).join(" ") === expando;

// Support: Chrome<14
// Always assume duplicates if they aren't passed to the comparison function
support.detectDuplicates = hasDuplicate;

// Initialize against the default document
setDocument();

// Support: Webkit<537.32 - Safari 6.0.3/Chrome 25 (fixed in Chrome 27)
// Detached nodes confoundingly follow *each other*
support.sortDetached = assert(function( div1 ) {
    // Should return 1, but returns 4 (following)
}

```

```

return div1.compareDocumentPosition( document.createElement("div") ) & 1;
});

// Support: IE<8
// Prevent attribute/property "interpolation"
// http://msdn.microsoft.com/en-us/library/ms536429%28VS.85%29.aspx
if ( !assert(function( div ) {
    div.innerHTML = "<a href='#'></a>";
    return div.firstChild.getAttribute("href") === "#" ;
}) ) {
    addHandle( "type[href|height|width]", function( elem, name, isXML ) {
        if ( !isXML ) {
            return elem.getAttribute( name, name.toLowerCase() === "type" ?
1 : 2 );
        }
    });
}

// Support: IE<9
// Use defaultValue in place of getAttribute("value")
if ( !support.attributes || !assert(function( div ) {
    div.innerHTML = "<input/>";
    div.firstChild.setAttribute( "value", "" );
    return div.firstChild.getAttribute( "value" ) === "" ;
}) ) {
    addHandle( "value", function( elem, name, isXML ) {
        if ( !isXML && elem.nodeName.toLowerCase() === "input" ) {
            return elem.defaultValue;
        }
    });
}

// Support: IE<9
// Use getAttributeNode to fetch booleans when getAttribute lies
if ( !assert(function( div ) {
    return div.getAttribute("disabled") == null;
}) ) {
    addHandle( booleans, function( elem, name, isXML ) {
        var val;
        if ( !isXML ) {
            return (val = elem.getAttributeNode( name )) && val.specified ?
                val.value :
                elem[ name ] === true ? name.toLowerCase() : null;
        }
    });
}

```

```

}

jQuery.find = Sizzle;
jQuery.expr = Sizzle.selectors;
jQuery.expr[":"] = jQuery.expr.pseudos;
jQuery.unique = Sizzle.uniqueSort;
jQuery.text = Sizzle.getText;
jQuery.isXMLDoc = Sizzle.isXML;
jQuery.contains = Sizzle.contains;

})( window );
// String to Object options format cache
var optionsCache = {};

// Convert String-formatted options into Object-formatted ones and store in cache
function createOptions( options ) {
    var object = optionsCache[ options ] = {};
    jQuery.each( options.match( core_rnotwhite ) || [], function( _, flag ) {
        object[ flag ] = true;
    });
    return object;
}

/*
 * Create a callback list using the following parameters:
 *
 *   options: an optional list of space-separated options that will change how
 *           the callback list behaves or a more traditional option object
 *
 * By default a callback list will act like an event callback list and can be
 * "fired" multiple times.
 *
 * Possible options:
 *
 *   once:          will ensure the callback list can only be fired once (like a
Deferred)
 *
 *   memory:        will keep track of previous values and will call any
callback added
 *
 *               after the list has been fired right away with the latest
"memorized"
 *
 *               values (like a Deferred)
 *

```

```

*      unique:           will ensure a callback can only be added once (no
duplicate in the list)
*
*      stopOnFalse:interrupt callings when a callback returns false
*
*/
jQuery.Callbacks = function( options ) {

    // Convert options from String-formatted to Object-formatted if needed
    // (we check in cache first)
    options = typeof options === "string" ?
        ( optionsCache[ options ] || createOptions( options ) ) :
        jQuery.extend( {}, options );

    var // Flag to know if list is currently firing
        firing,
        // Last fire value (for non-forgettable lists)
        memory,
        // Flag to know if list was already fired
        fired,
        // End of the loop when firing
        firingLength,
        // Index of currently firing callback (modified by remove if needed)
        firingIndex,
        // First callback to fire (used internally by add and fireWith)
        firingStart,
        // Actual callback list
        list = [],
        // Stack of fire calls for repeatable lists
        stack = !options.once && [],
        // Fire callbacks
        fire = function( data ) {
            memory = options.memory && data;
            fired = true;
            firingIndex = firingStart || 0;
            firingStart = 0;
            firingLength = list.length;
            firing = true;
            for ( ; list && firingIndex < firingLength; firingIndex++ ) {
                if ( list[ firingIndex ].apply( data[ 0 ], data[ 1 ] ) === false &&
options.stopOnFalse ) {
                    memory = false; // To prevent further calls using add
break;
                }
            }
        }
}

```

```

firing = false;
if ( list ) {
    if ( stack ) {
        if ( stack.length ) {
            fire( stack.shift() );
        }
    } else if ( memory ) {
        list = [];
    } else {
        self.disable();
    }
},
// Actual Callbacks object
self = {
    // Add a callback or a collection of callbacks to the list
    add: function() {
        if ( list ) {
            // First, we save the current length
            var start = list.length;
            (function add( args ) {
                jQuery.each( args, function( _, arg ) {
                    var type = jQuery.type( arg );
                    if ( type === "function" ) {
                        if ( !options.unique || !self.has( arg ) )
{
                            list.push( arg );
                        }
                    } else if ( arg && arg.length && type !==
"string" ) {
                            // Inspect recursively
                            add( arg );
                        }
                    });
                })( arguments );
            // Do we need to add the callbacks to the
            // current firing batch?
            if ( firing ) {
                firingLength = list.length;
                // With memory, if we're not firing then
                // we should call right away
            } else if ( memory ) {
                firingStart = start;
                fire( memory );
            }
        }
    }
}

```

```

        }
        return this;
    },
    // Remove a callback from the list
    remove: function() {
        if ( list ) {
            jQuery.each( arguments, function( _, arg ) {
                var index;
                while( ( index = jQuery.inArray( arg, list, index ) ) )
> -1 ) {
                    list.splice( index, 1 );
                    // Handle firing indexes
                    if ( firing ) {
                        if ( index <= firingLength ) {
                            firingLength--;
                        }
                        if ( index <= firingIndex ) {
                            firingIndex--;
                        }
                    }
                }
            });
        }
        return this;
    },
    // Check if a given callback is in the list.
    // If no argument is given, return whether or not list has callbacks
attached.
    has: function( fn ) {
        return fn ? jQuery.inArray( fn, list ) > -1 : !( list && list.length );
    },
    // Remove all callbacks from the list
    empty: function() {
        list = [];
        firingLength = 0;
        return this;
    },
    // Have the list do nothing anymore
    disable: function() {
        list = stack = memory = undefined;
        return this;
    },
    // Is it disabled?
    disabled: function() {
        return !list;
    }
}

```

```

        },
        // Lock the list in its current state
        lock: function() {
            stack = undefined;
            if ( !memory ) {
                self.disable();
            }
            return this;
        },
        // Is it locked?
        locked: function() {
            return !stack;
        },
        // Call all callbacks with the given context and arguments
        fireWith: function( context, args ) {
            if ( list && ( !fired || stack ) ) {
                args = args || [];
                args = [ context, args.slice ? args.slice() : args ];
                if ( firing ) {
                    stack.push( args );
                } else {
                    fire( args );
                }
            }
            return this;
        },
        // Call all the callbacks with the given arguments
        fire: function() {
            self.fireWith( this, arguments );
            return this;
        },
        // To know if the callbacks have already been called at least once
        fired: function() {
            return !!fired;
        }
    );
};

return self;
};

jQuery.extend({
    Deferred: function( func ) {
        var tuples = [
            // action, add listener, listener list, final state

```

```

[ "resolve", "done", jQuery.Callbacks("once memory"),
"resolved" ],
[ "reject", "fail", jQuery.Callbacks("once memory"),
"rejected" ],
[ "notify", "progress", jQuery.Callbacks("memory") ]
],
state = "pending",
promise = {
    state: function() {
        return state;
    },
    always: function() {
        deferred.done( arguments ).fail( arguments );
        return this;
    },
    then: function( /* fnDone, fnFail, fnProgress */ ) {
        var fns = arguments;
        return jQuery.Deferred(function( newDefer ) {
            jQuery.each( tuples, function( i, tuple ) {
                var action = tuple[ 0 ],
                    fn = jQueryisFunction( fns[ i ] ) &&
fn[ i ];
                // deferred[ done | fail | progress ] for
forwarding actions to newDefer
                deferred[ tuple[1] ](function() {
                    var returned = fn && fn.apply( this,
arguments );
                    if ( returned &&
jQueryisFunction( returned.promise ) ) {
                        returned.promise()
                            .done( newDefer.resolve )
                            .fail( newDefer.reject )
                            .progress( newDefer.notify );
                    } else {
                        newDefer[ action + "With" ]
( this === promise ? newDefer.promise() : this, fn ? [ returned ] : arguments );
                    }
                });
            });
            fns = null;
        }).promise();
    },
    // Get a promise for this deferred

```

```

// If obj is provided, the promise aspect is added to the object
promise: function( obj ) {
    return obj != null ? jQuery.extend( obj, promise ) :
promise;
}
},
deferred = {};

// Keep pipe for back-compat
promise.pipe = promise.then;

// Add list-specific methods
jQuery.each( tuples, function( i, tuple ) {
    var list = tuple[ 2 ],
        stateString = tuple[ 3 ];

    // promise[ done | fail | progress ] = list.add
    promise[ tuple[1] ] = list.add;

    // Handle state
    if ( stateString ) {
        list.add(function() {
            // state = [ resolved | rejected ]
            state = stateString;

            // [ reject_list | resolve_list ].disable; progress_list.lock
            }, tuples[ i ^ 1 ][ 2 ].disable, tuples[ 2 ][ 2 ].lock );
    }

    // deferred[ resolve | reject | notify ]
    deferred[ tuple[0] ] = function() {
        deferred[ tuple[0] + "With" ]( this === deferred ? promise :
this, arguments );
        return this;
    };
    deferred[ tuple[0] + "With" ] = list.fireWith;
});

// Make the deferred a promise
promise.promise( deferred );

// Call given func if any
if ( func ) {
    func.call( deferred, deferred );
}

```

```

// All done!
return deferred;
},

// Deferred helper
when: function( subordinate /* , ..., subordinateN */ ) {
    var i = 0,
        resolveValues = core_slice.call( arguments ),
        length = resolveValues.length,

        // the count of uncompleted subordinates
        remaining = length !== 1 || ( subordinate &&
jQueryisFunction( subordinate.promise ) ) ? length : 0,
        // the master Deferred. If resolveValues consist of only a single
Deferred, just use that.
        deferred = remaining === 1 ? subordinate : jQuery.Deferred(),

        // Update function for both resolve and progress values
        updateFunc = function( i, contexts, values ) {
            return function( value ) {
                contexts[ i ] = this;
                values[ i ] = arguments.length > 1 ?
core_slice.call( arguments ) : value;
                if( values === progressValues ) {
                    deferred.notifyWith( contexts, values );
                } else if ( !( --remaining ) ) {
                    deferred.resolveWith( contexts, values );
                }
            };
        },
        progressValues, progressContexts, resolveContexts;

        // add listeners to Deferred subordinates; treat others as resolved
        if ( length > 1 ) {
            progressValues = new Array( length );
            progressContexts = new Array( length );
            resolveContexts = new Array( length );
            for ( ; i < length; i++ ) {
                if ( resolveValues[ i ] &&
jQueryisFunction( resolveValues[ i ].promise ) ) {
                    resolveValues[ i ].promise()
                }
            }
        }
    }
}

```

```

        .done( updateFunc( i, resolveContexts,
resolveValues ) )
            .fail( deferred.reject )
            .progress( updateFunc( i, progressContexts,
progressValues ) );
    } else {
        --remaining;
    }
}
}

// if we're not waiting on anything, resolve the master
if ( !remaining ) {
    deferred.resolveWith( resolveContexts, resolveValues );
}

return deferred.promise();
}
});

jQuery.support = (function( support ) {

var all, a, input, select, fragment, opt, eventName, isSupported, i,
    div = document.createElement("div");

// Setup
div.setAttribute( "className" , "t" );
div.innerHTML = " <link><table></table><a href='/a'>a</a><input
type='checkbox' />";

// Finish early in limited (non-browser) environments
all = div.getElementsByTagName("*") || [];
a = div.getElementsByTagName("a")[ 0 ];
if ( !a || !a.style || !all.length ) {
    return support;
}

// First batch of tests
select = document.createElement("select");
opt = select.appendChild( document.createElement("option" ) );
input = div.getElementsByTagName("input")[ 0 ];

a.style.cssText = "top:1px;float:left;opacity:.5";

// Test setAttribute on camelCase class. If it works, we need attrFixes when doing
get/setAttribute (ie6/7)

```

```
support.getSetAttribute = div.className !== "t";

// IE strips leading whitespace when .innerHTML is used
support.leadingWhitespace = div.firstChild.nodeType === 3;

// Make sure that tbody elements aren't automatically inserted
// IE will insert them into empty tables
support.tbody = !div.getElementsByTagName("tbody").length;

// Make sure that link elements get serialized correctly by innerHTML
// This requires a wrapper element in IE
support.htmlSerialize = !!div.getElementsByTagName("link").length;

// Get the style information from getAttribute
// (IE uses .cssText instead)
support.style = /top/.test( a.getAttribute("style") );

// Make sure that URLs aren't manipulated
// (IE normalizes it by default)
support.hrefNormalized = a.getAttribute("href") === "/a";

// Make sure that element opacity exists
// (IE uses filter instead)
// Use a regex to work around a WebKit issue. See #5145
support.opacity = /^0\.5/.test( a.style.opacity );

// Verify style float existence
// (IE uses styleFloat instead of cssFloat)
support.cssFloat = !!a.style.cssFloat;

// Check the default checkbox/radio value (" on WebKit; "on" elsewhere)
support.checkOn = !!input.value;

// Make sure that a selected-by-default option has a working selected property.
// (WebKit defaults to false instead of true, IE too, if it's in an optgroup)
support.optSelected = opt.selected;

// Tests for enctype support on a form (#6743)
support.enctype = !!document.createElement("form").enctype;

// Makes sure cloning an html5 element does not cause problems
// Where outerHTML is undefined, this still works
support.html5Clone =
document.createElement("nav").cloneNode( true ).outerHTML !== "<:nav></:nav>";
```

```
// Will be defined later
support.inlineBlockNeedsLayout = false;
support.shrinkWrapBlocks = false;
support.pixelPosition = false;
support.deleteExpando = true;
support.noCloneEvent = true;
support.reliableMarginRight = true;
support.boxSizingReliable = true;

// Make sure checked status is properly cloned
input.checked = true;
support.noCloneChecked = input.cloneNode( true ).checked;

// Make sure that the options inside disabled selects aren't marked as disabled
// (WebKit marks them as disabled)
select.disabled = true;
support.optDisabled = !opt.disabled;

// Support: IE<9
try {
    delete div.test;
} catch( e ) {
    support.deleteExpando = false;
}

// Check if we can trust getAttribute("value")
input = document.createElement("input");
input.setAttribute( "value", "" );
support.input = input.getAttribute( "value" ) === "";

// Check if an input maintains its value after becoming a radio
input.value = "t";
input.setAttribute( "type", "radio" );
support.radioValue = input.value === "t";

// #11217 - WebKit loses check when the name is after the checked attribute
input.setAttribute( "checked", "t" );
input.setAttribute( "name", "t" );

fragment = document.createDocumentFragment();
fragment.appendChild( input );

// Check if a disconnected checkbox will retain its checked
// value of true after appended to the DOM (IE6/7)
support.appendChecked = input.checked;
```

```

// WebKit doesn't clone checked state correctly in fragments
support.checkClone =
fragment.cloneNode( true ).cloneNode( true ).lastChild.checked;

// Support: IE<9
// Opera does not clone events (and typeof div.attachEvent === undefined).
// IE9-10 clones events bound via attachEvent, but they don't trigger with .click()
if ( div.attachEvent ) {
    div.attachEvent( "onclick", function() {
        support.noCloneEvent = false;
    });
}

div.cloneNode( true ).click();
}

// Support: IE<9 (lack submit/change bubble), Firefox 17+ (lack focusin event)
// Beware of CSP restrictions (https://developer.mozilla.org/en/Security/CSP)
for ( i in { submit: true, change: true, focusin: true } ) {
    div.setAttribute( eventName = "on" + i, "t" );

    support[ i + "Bubbles" ] = eventName in window ||
div.attributes[ eventName ].expando === false;
}

div.style.backgroundClip = "content-box";
div.cloneNode( true ).style.backgroundClip = " ";
support.clearCloneStyle = div.style.backgroundClip === "content-box";

// Support: IE<9
// Iteration over object's inherited properties before its own.
for ( i in jQuery( support ) ) {
    break;
}
support.ownLast = i !== "0";

// Run tests that need a body at doc ready
jQuery(function() {
    var container, marginDiv, tds,
        divReset = "padding:0;margin:0;border:0;display:block;box-
sizing:content-box;-moz-box-sizing:content-box;-webkit-box-sizing:content-box;",
        body = document.getElementsByTagName("body")[0];

    if ( !body ) {
        // Return for frameset docs that don't have a body

```

```

        return;
    }

    container = document.createElement("div");
    container.style.cssText = "border:0;width:0;height:0;position:absolute;top:0;left:-9999px;margin-top:1px";

    body.appendChild( container ).appendChild( div );

    // Support: IE8
    // Check if table cells still have offsetWidth/Height when they are set
    // to display:none and there are still other visible table cells in a
    // table row; if so, offsetWidth/Height are not reliable for use when
    // determining if an element has been hidden directly using
    // display:none (it is still safe to use offsets if a parent element is
    // hidden; don safety goggles and see bug #4512 for more information).
    div.innerHTML = "<table><tr><td></td><td>t</td></tr></table>";
    tds = div.getElementsByTagName("td");
    tds[ 0 ].style.cssText = "padding:0;margin:0;border:0;display:none";
    isSupported = ( tds[ 0 ].offsetHeight === 0 );

    tds[ 0 ].style.display = " ";
    tds[ 1 ].style.display = "none";

    // Support: IE8
    // Check if empty table cells still have offsetWidth/Height
    support.reliableHiddenOffsets = isSupported && ( tds[ 0 ].offsetHeight ===
0 );

    // Check box-sizing and margin behavior.
    div.innerHTML = "";
    div.style.cssText = "box-sizing:border-box;-moz-box-sizing:border-box;-webkit-box-sizing:border-box;padding:1px;border:1px;display:block;width:4px;margin-top:1%;position:absolute;top:1%;";

    // Workaround failing boxSizing test due to offsetWidth returning wrong
    value
    // with some non-1 values of body zoom, ticket #13543
    jQuery.swap( body, body.style.zoom != null ? { zoom: 1 } : {}, function() {
        support.boxSizing = div.offsetWidth === 4;
    });

    // Use window.getComputedStyle because jsdom on node.js will break
    without it.
    if ( window.getComputedStyle ) {

```

```

support.pixelPosition = ( window.getComputedStyle( div, null ) ||
{} ).top !== "1%";
support.boxSizingReliable = ( window.getComputedStyle( div, null ) ||
{ width: "4px" } ).width === "4px";

// Check if div with explicit width and no margin-right incorrectly
// gets computed margin-right based on width of container. (#3333)
// Fails in WebKit before Feb 2011 nightlies
// WebKit Bug 13343 - getComputedStyle returns wrong value for
margin-right
marginDiv = div.appendChild( document.createElement("div") );
marginDiv.style.cssText = div.style.cssText = divReset;
marginDiv.style.marginRight = marginDiv.style.width = "0";
div.style.width = "1px";

support.reliableMarginRight =
!parseFloat( ( window.getComputedStyle( marginDiv, null ) ||
{} ).marginRight );
}

if ( typeof div.style.zoom !== core_undefined ) {
// Support: IE<8
// Check if natively block-level elements act like inline-block
// elements when setting their display to 'inline' and giving
// them layout
div.innerHTML = "";
div.style.cssText = divReset + "width:1px;padding:
1px;display:inline;zoom:1";
support.inlineBlockNeedsLayout = ( div.offsetWidth === 3 );

// Support: IE6
// Check if elements with layout shrink-wrap their children
div.style.display = "block";
div.innerHTML = "<div></div>";
div.firstChild.style.width = "5px";
support.shrinkWrapBlocks = ( div.offsetWidth !== 3 );

if ( support.inlineBlockNeedsLayout ) {
// Prevent IE 6 from affecting layout for positioned elements
#11048
// Prevent IE from shrinking the body in IE 7 mode #12869
// Support: IE<8
body.style.zoom = 1;
}
}

```

```

        body.removeChild( container );

        // Null elements to avoid leaks in IE
        container = div = tds = marginDiv = null;
    });

    // Null elements to avoid leaks in IE
    all = select = fragment = opt = a = input = null;

    return support;
})( {});

var rbrace = /(?:\{[\s\S]*\}\|[\s\S]*\})$/,
    rmultiDash = /([A-Z])/g;

function internalData( elem, name, data, pvt /* Internal Use Only */ ){
    if ( !jQuery.acceptData( elem ) ) {
        return;
    }

    var ret, thisCache,
        internalKey = jQuery.expando,

        // We have to handle DOM nodes and JS objects differently because IE6-7
        // can't GC object references properly across the DOM-JS boundary
        isNode = elem.nodeType,

        // Only DOM nodes need the global jQuery cache; JS object data is
        // attached directly to the object so GC can occur automatically
        cache = isNode ? jQuery.cache : elem,

        // Only defining an ID for JS objects if its cache already exists allows
        // the code to shortcut on the same path as a DOM node with no cache
        id = isNode ? elem[ internalKey ] : elem[ internalKey ] && internalKey;

    // Avoid doing any more work than we need to when trying to get data on an
    // object that has no data at all
    if ( (!id || !cache[id] || (!pvt && !cache[id].data)) && data === undefined && typeof
name === "string" ){
        return;
    }

    if ( !id ){
        // Only DOM nodes need a new unique ID for each element since their data

```

```

// ends up in the global cache
if ( isNode ) {
    id = elem[ internalKey ] = core_deletedIds.pop() || jQuery.guid++;
} else {
    id = internalKey;
}
}

if ( !cache[ id ] ) {
    // Avoid exposing jQuery metadata on plain JS objects when the object
    // is serialized using JSON.stringify
    cache[ id ] = isNode ? {} : { toJSON: jQuery.noop };
}

// An object can be passed to jQuery.data instead of a key/value pair; this gets
// shallow copied over onto the existing cache
if ( typeof name === "object" || typeof name === "function" ) {
    if ( pvt ) {
        cache[ id ] = jQuery.extend( cache[ id ], name );
    } else {
        cache[ id ].data = jQuery.extend( cache[ id ].data, name );
    }
}

thisCache = cache[ id ];

// jQuery data() is stored in a separate object inside the object's internal data
// cache in order to avoid key collisions between internal data and user-defined
// data.
if ( !pvt ) {
    if ( !thisCache.data ) {
        thisCache.data = {};
    }

    thisCache = thisCache.data;
}

if ( data !== undefined ) {
    thisCache[ jQuery.camelCase( name ) ] = data;
}

// Check for both converted-to-camel and non-converted data property names
// If a data property was specified
if ( typeof name === "string" ) {

```

```

// First Try to find as-is property data
ret = thisCache[ name ];

// Test for nullundefined property data
if ( ret == null ) {

    // Try to find the camelCased property
    ret = thisCache[ jQuery.camelCase( name ) ];
}
} else {
    ret = thisCache;
}

return ret;
}

function internalRemoveData( elem, name, pvt ) {
    if ( !jQuery.acceptData( elem ) ) {
        return;
    }

    var thisCache, i,
        isNode = elem.nodeType,

        // See jQuery.data for more information
        cache = isNode ? jQuery.cache : elem,
        id = isNode ? elem[ jQuery.expando ] : jQuery.expando;

    // If there is already no cache entry for this object, there is no
    // purpose in continuing
    if ( !cache[ id ] ) {
        return;
    }

    if ( name ) {

        thisCache = pvt ? cache[ id ] : cache[ id ].data;

        if ( thisCache ) {

            // Support array or space separated string names for data keys
            if ( !jQuery.isArray( name ) ) {

                // try the string as a key before any manipulation
                if ( name in thisCache ) {

```

```

        name = [ name ];
    } else {

        // split the camel cased version by spaces unless a key
with the spaces exists

        name = jQuery.camelCase( name );
        if ( name in thisCache ) {
            name = [ name ];
        } else {
            name = name.split(" ");
        }
    } else {
        // If "name" is an array of keys...
        // When data is initially created, via ("key", "val") signature,
        // keys will be converted to camelCase.
        // Since there is no way to tell _how_ a key was added, remove
        // both plain key and camelCase key. #12786
        // This will only penalize the array argument path.
        name = name.concat( jQuery.map( name, jQuery.camelCase ) );
    }

    i = name.length;
    while ( i-- ) {
        delete thisCache[ name[i] ];
    }

    // If there is no data left in the cache, we want to continue
    // and let the cache object itself get destroyed
    if ( pvt ? !isEmptyDataObject(thisCache) : !
jQuery.isEmptyObject(thisCache) ) {
        return;
    }
}

// See jQuery.data for more information
if ( !pvt ) {
    delete cache[ id ].data;

    // Don't destroy the parent cache unless the internal data object
    // had been the only thing left in it
    if ( !isEmptyDataObject( cache[ id ] ) ) {
        return;
    }
}

```

```

}

// Destroy the cache
if ( isNode ) {
    jQuery.cleanData( [ elem ], true );

    // Use delete when supported for expandos or `cache` is not a window per
isWindow (#10080)
    /* jshint eqeqeq: false */
} else if ( jQuery.support.deleteExpando || cache != cache.window ) {
    /* jshint eqeqeq: true */
    delete cache[ id ];

    // When all else fails, null
} else {
    cache[ id ] = null;
}
}

jQuery.extend({
    cache: {},

    // The following elements throw uncatchable exceptions if you
    // attempt to add expando properties to them.
    noData: {
        "applet": true,
        "embed": true,
        // Ban all objects except for Flash (which handle expandos)
        "object": "clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
    },
    hasData: function( elem ) {
        elem = elem.nodeType ? jQuery.cache[ elem[jQuery.expando] ] :
elem[ jQuery.expando ];
        return !!elem && !isEmptyDataObject( elem );
    },
    data: function( elem, name, data ) {
        return internalData( elem, name, data );
    },
    removeData: function( elem, name ) {
        return internalRemoveData( elem, name );
    },
}
);

```

```

// For internal use only.
_data: function( elem, name, data ) {
    return internalData( elem, name, data, true );
},

_removeData: function( elem, name ) {
    return internalRemoveData( elem, name, true );
},

// A method for determining if a DOM node can handle the data expando
acceptData: function( elem ) {
    // Do not set data on non-element because it will not be cleared (#8335).
    if ( elem.nodeType && elem.nodeType !== 1 && elem.nodeType !== 9 ) {
        return false;
    }

    var noData = elem.nodeName &&
jQuery.noData[ elem.nodeName.toLowerCase() ];

    // nodes accept data unless otherwise specified; rejection can be conditional
    return !noData || noData !== true && elem.getAttribute("classid") ===
noData;
}
});

jQuery.fn.extend({
    data: function( key, value ) {
        var attrs, name,
            data = null,
            i = 0,
            elem = this[0];

        // Special exceptions of .data basically thwart jQuery.access,
        // so implement the relevant behavior ourselves

        // Gets all values
        if ( key === undefined ) {
            if ( this.length ) {
                data = jQuery.data( elem );

                if ( elem.nodeType === 1 && !jQuery._data( elem,
"parsedAttrs" ) ) {
                    attrs = elem.attributes;
                    for ( ; i < attrs.length; i++ ) {
                        name = attrs[i].name;

```

```

        if ( name.indexOf("data-") === 0 ) {
            name = jQuery.camelCase( name.slice(5) );

                dataAttr( elem, name, data[ name ] );
            }
        }
    }

    return data;
}

// Sets multiple values
if ( typeof key === "object" ) {
    return this.each(function() {
        jQuery.data( this, key );
    });
}

return arguments.length > 1 ?

    // Sets one value
    this.each(function() {
        jQuery.data( this, key, value );
    }) :

    // Gets one value
    // Try to fetch any internally stored data first
    elem ? dataAttr( elem, key, jQuery.data( elem, key ) ) : null;
},

removeData: function( key ) {
    return this.each(function() {
        jQuery.removeData( this, key );
    });
}
};

function dataAttr( elem, key, data ) {
    // If nothing was found internally, try to fetch any
    // data from the HTML5 data-* attribute
    if ( data === undefined && elem.nodeType === 1 ) {

```

```

var name = "data-" + key.replace( rmultiDash, "-$1" ).toLowerCase();

data = elem.getAttribute( name );

if ( typeof data === "string" ) {
    try {
        data = data === "true" ? true :
            data === "false" ? false :
            data === "null" ? null :
                // Only convert to a number if it doesn't change the
                string
                +data + "" === data ? +data :
                    rbrace.test( data ) ? jQuery.parseJSON( data ) :
                        data;
    } catch( e ) {}

        // Make sure we set the data so it isn't changed later
        jQuery.data( elem, key, data );
    }

} else {
    data = undefined;
}
}

return data;
}

// checks a cache object for emptiness
function isEmptyDataObject( obj ) {
    var name;
    for ( name in obj ) {

        // if the public data object is empty, the private is still empty
        if ( name === "data" && jQuery.isEmptyObject( obj[name] ) ) {
            continue;
        }
        if ( name !== "toJSON" ) {
            return false;
        }
    }

    return true;
}
jQuery.extend({
    queue: function( elem, type, data ) {

```

```

var queue;

if ( elem ) {
    type = ( type || "fx" ) + "queue";
    queue = jQuery._data( elem, type );

    // Speed up dequeue by getting out quickly if this is just a lookup
    if ( data ) {
        if ( !queue || jQuery.isArray(data) ) {
            queue = jQuery._data( elem, type,
jQuery.makeArray(data) );
        } else {
            queue.push( data );
        }
    }
    return queue || [];
}

},
dequeue: function( elem, type ) {
    type = type || "fx";

    var queue = jQuery.queue( elem, type ),
        startLength = queue.length,
        fn = queue.shift(),
        hooks = jQuery._queueHooks( elem, type ),
        next = function() {
            jQuery.dequeue( elem, type );
        };

    // If the fx queue is dequeued, always remove the progress sentinel
    if ( fn === "inprogress" ) {
        fn = queue.shift();
        startLength--;
    }

    if ( fn ) {

        // Add a progress sentinel to prevent the fx queue from being
        // automatically dequeued
        if ( type === "fx" ) {
            queue.unshift( "inprogress" );
        }

        // clear up the last queue stop function
    }
}

```

```

        delete hooks.stop;
        fn.call( elem, next, hooks );
    }

    if ( !startLength && hooks ) {
        hooks.empty.fire();
    }
},

// not intended for public consumption - generates a queueHooks object, or
returns the current one
_queueHooks: function( elem, type ) {
    var key = type + "queueHooks";
    return jQuery._data( elem, key ) || jQuery._data( elem, key, {
        empty: jQuery.Callbacks("once memory").add(function() {
            jQuery._removeData( elem, type + "queue" );
            jQuery._removeData( elem, key );
        })
    });
}

jQuery.fn.extend({
    queue: function( type, data ) {
        var setter = 2;

        if ( typeof type !== "string" ) {
            data = type;
            type = "fx";
            setter--;
        }

        if ( arguments.length < setter ) {
            return jQuery.queue( this[0], type );
        }

        return data === undefined ?
            this :
            this.each(function() {
                var queue = jQuery.queue( this, type, data );

                // ensure a hooks for this queue
                jQuery._queueHooks( this, type );

                if ( type === "fx" && queue[0] !== "inprogress" ) {

```

```

        jQuery.dequeue( this, type );
    }
});

},
dequeue: function( type ) {
    return this.each(function() {
        jQuery.dequeue( this, type );
    });
},
// Based off of the plugin by Clint Helfers, with permission.
// http://blindsignals.com/index.php/2009/07/jquery-delay/
delay: function( time, type ) {
    time = jQuery.fx ? jQuery.fx.speeds[ time ] || time : time;
    type = type || "fx";

    return this.queue( type, function( next, hooks ) {
        var timeout = setTimeout( next, time );
        hooks.stop = function() {
            clearTimeout( timeout );
        };
    });
},
clearQueue: function( type ) {
    return this.queue( type || "fx", [] );
},
// Get a promise resolved when queues of a certain type
// are emptied (fx is the type by default)
promise: function( type, obj ) {
    var tmp,
        count = 1,
        defer = jQuery.Deferred(),
        elements = this,
        i = this.length,
        resolve = function() {
            if ( !( --count ) ) {
                defer.resolveWith( elements, [ elements ] );
            }
        };
    if ( typeof type !== "string" ) {
        obj = type;
        type = undefined;
    }
    type = type || "fx";
}

```

```

        while( i-- ) {
            tmp = jQuery._data( elements[ i ], type + "queueHooks" );
            if ( tmp && tmp.empty ) {
                count++;
                tmp.empty.add( resolve );
            }
        }
        resolve();
        return defer.promise( obj );
    }
});

var nodeHook, boolHook,
rclass = /[ \t\r\n\f]/g,
rreturn = /\r/g,
rfocusable = /^(:input|select|text|area|button|object)$|/i,
rclickable = /^(:checkbox|radio)$|/i,
ruseDefault = /^(:checked|selected)$|/i,
getSetAttribute = jQuery.support.getSetAttribute,
getSetInput = jQuery.support.input;

jQuery.fn.extend({
    attr: function( name, value ) {
        return jQuery.access( this, jQuery.attr, name, value, arguments.length > 1 );
    },

    removeAttr: function( name ) {
        return this.each(function() {
            jQuery.removeAttr( this, name );
        });
    },

    prop: function( name, value ) {
        return jQuery.access( this, jQuery.prop, name, value, arguments.length > 1 );
    },

    removeProp: function( name ) {
        name = jQuery.propFix[ name ] || name;
        return this.each(function() {
            // try/catch handles cases where IE balks (such as removing a
            // property on window)
            try {
                this[ name ] = undefined;
                delete this[ name ];
            } catch( e ) {}
        });
    }
});

```

```

    },
    addClass: function( value ) {
        var classes, elem, cur, clazz, j,
            i = 0,
            len = this.length,
            proceed = typeof value === "string" && value;

        if ( jQueryisFunction( value ) ) {
            return this.each(function( j ) {
                jQuery( this ).addClass( value.call( this, j, this.className ) );
            });
        }

        if ( proceed ) {
            // The disjunction here is for better compressibility (see removeClass)
            classes = ( value || "" ).match( core_rnotwhite ) || [];

            for ( ; i < len; i++ ) {
                elem = this[ i ];
                cur = elem.nodeType === 1 && ( elem.className ?
                    ( " " + elem.className + " " ).replace( rclass, " " ) :
                    " " );
                );

                if ( cur ) {
                    j = 0;
                    while ( (clazz = classes[j++]) ) {
                        if ( cur.indexOf( " " + clazz + " " ) < 0 ) {
                            cur += clazz + " ";
                        }
                    }
                    elem.className = jQuery.trim( cur );
                }
            }
        }
    }

    return this;
},
removeClass: function( value ) {
    var classes, elem, cur, clazz, j,
        i = 0,
        len = this.length,

```

```

proceed = arguments.length === 0 || typeof value === "string" &&
value;

if ( jQueryisFunction( value ) ) {
    return this.each(function( j ) {
        jQuery( this ).removeClass( value.call( this, j, this.className ) );
    });
}

if ( proceed ) {
    classes = ( value || "" ).match( core_rnotwhite ) || [];

    for ( ; i < len; i++ ) {
        elem = this[ i ];
        // This expression is here for better compressibility (see
addClass)
        cur = elem.nodeType === 1 && ( elem.className ?
            ( " " + elem.className + " " ).replace( rclass, " " ) :
            ""
        );

        if ( cur ) {
            j = 0;
            while ( (clazz = classes[j++]) ) {
                // Remove *all* instances
                while ( cur.indexOf( " " + clazz + " " ) >= 0 ) {
                    cur = cur.replace( " " + clazz + " ", " " );
                }
            }
            elem.className = value ? jQuery.trim( cur ) : "";
        }
    }
}

return this;
},

toggleClass: function( value, stateVal ) {
    var type = typeof value;

    if ( typeof stateVal === "boolean" && type === "string" ) {
        return stateVal ? this.addClass( value ) : this.removeClass( value );
    }

    if ( jQueryisFunction( value ) ) {
        return this.each(function( i ) {

```

```

jQuery( this ).toggleClass( value.call(this, i, this.className,
stateVal), stateVal );
    });
}

return this.each(function() {
    if ( type === "string" ) {
        // toggle individual class names
        var className,
            i = 0,
            self = jQuery( this ),
            classNames = value.match( core_rnotwhite ) || [];
        while ( (className = classNames[ i++ ]) ) {
            // check each className given, space separated list
            if ( self.hasClass( className ) ) {
                self.removeClass( className );
            } else {
                self.addClass( className );
            }
        }
        // Toggle whole class name
    } else if ( type === core_undefined || type === "boolean" ) {
        if ( this.className ) {
            // store className if set
            jQuery._data( this, "__className__", this.className );
        }
        // If the element has a class name or if we're passed "false",
        // then remove the classname (if there was one, the
above saved it).
        // Otherwise bring back whatever was previously saved (if
anything),
        // falling back to the empty string if nothing was stored.
        this.className = this.className || value === false ? "" :
jQuery._data( this, "__className__" ) || "";
    }
},
hasClass: function( selector ) {
    var className = " " + selector + " ",
        i = 0,
        l = this.length;
}
);
}

```

```

for ( ; i < l; i++ ) {
    if ( this[i].nodeType === 1 && (" " + this[i].className + "
").replace(rclass, " ").indexOf( className ) >= 0 ) {
        return true;
    }
}

return false;
},

val: function( value ) {
    var ret, hooks,isFunction,
        elem = this[0];

    if ( !arguments.length ) {
        if ( elem ) {
            hooks = jQuery.valHooks[ elem.type ] ||
jQuery.valHooks[ elem.nodeName.toLowerCase() ];

            if ( hooks && "get" in hooks && (ret = hooks.get( elem,
"value" )) !== undefined ) {
                return ret;
            }
        }

        ret = elem.value;

        return typeof ret === "string" ?
            // handle most common string cases
            ret.replace(rreturn, "") :
            // handle cases where value is null/undef or number
            ret == null ? "" : ret;
    }

    return;
}

isFunction = jQuery.isFunction( value );

return this.each(function( i ) {
    var val;

    if ( this.nodeType !== 1 ) {
        return;
    }
}

```

```

        if (isFunction) {
            val = value.call( this, i, jQuery(this).val() );
        } else {
            val = value;
        }

        // Treat null/undefined as ""; convert numbers to string
        if (val == null) {
            val = "";
        } else if (typeof val === "number") {
            val += "";
        } else if (jQuery.isArray(val)) {
            val = jQuery.map(val, function (value) {
                return value == null ? "" : value + "";
            });
        }
    }

    hooks = jQuery.valHooks[ this.type ] ||
jQuery.valHooks[ this.nodeName.toLowerCase() ];

    // If set returns undefined, fall back to normal setting
    if ( !hooks || !( "set" in hooks ) || hooks.set( this, val, "value" ) ===
undefined ) {
        this.value = val;
    }
};

jQuery.extend({
    valHooks: {
        option: {
            get: function( elem ) {
                // Use proper attribute retrieval(#6932, #12072)
                var val = jQuery.find.attr( elem, "value" );
                return val != null ?
                    val :
                    elem.text;
            }
        },
        select: {
            get: function( elem ) {
                var value, option,
                    options = elem.options,
                    index = elem.selectedIndex,

```

```

one = elem.type === "select-one" || index < 0,
values = one ? null : [],
max = one ? index + 1 : options.length,
i = index < 0 ?
    max :
    one ? index : 0;

// Loop through all the selected options
for ( ; i < max; i++ ) {
    option = options[ i ];

    // oldIE doesn't update selected after form reset (#2551)
    if ( ( option.selected || i === index ) &&
        // Don't return options that are disabled or
in a disabled optgroup
        ( !jQuery.support.optDisabled ? !
option.disabled : option.getAttribute("disabled") === null ) &&
        ( !option.parentNode.disabled || !
jQuery.nodeName( option.parentNode, "optgroup" ) ) ) {

        // Get the specific value for the option
        value = jQuery( option ).val();

        // We don't need an array for one selects
        if ( one ) {
            return value;
        }

        // Multi-Selects return an array
        values.push( value );
    }
}

return values;
},

set: function( elem, value ) {
    var optionSet, option,
        options = elem.options,
        values = jQuery.makeArray( value ),
        i = options.length;

    while ( i-- ) {
        option = options[ i ];

```

```

        if ( (option.selected =
jQuery.inArray( jQuery(option).val(), values ) >= 0) ) {
            optionSet = true;
        }
    }

    // force browsers to behave consistently when non-matching
value is set
    if ( !optionSet ) {
        elem.selectedIndex = -1;
    }
    return values;
}
},
attr: function( elem, name, value ) {
    var hooks, ret,
        nType = elem.nodeType;

    // don't get/set attributes on text, comment and attribute nodes
    if ( !elem || nType === 3 || nType === 8 || nType === 2 ) {
        return;
    }

    // Fallback to prop when attributes are not supported
    if ( typeof elem.getAttribute === core_undefined ) {
        return jQuery.prop( elem, name, value );
    }

    // All attributes are lowercase
    // Grab necessary hook if one is defined
    if ( nType !== 1 || !jQuery.isXMLDoc( elem ) ) {
        name = name.toLowerCase();
        hooks = jQuery.attrHooks[ name ] ||
            ( jQuery.expr.match.bool.test( name ) ? boolHook :
nodeHook );
    }

    if ( value !== undefined ) {

        if ( value === null ) {
            jQueryremoveAttr( elem, name );
        }
    }
}

```

```

        } else if ( hooks && "set" in hooks && (ret = hooks.set( elem, value,
name )) !== undefined ) {
            return ret;

        } else {
            elem.setAttribute( name, value + " " );
            return value;
        }

    } else if ( hooks && "get" in hooks && (ret = hooks.get( elem, name )) !==
null ) {
        return ret;

    } else {
        ret = jQuery.find.attr( elem, name );

        // Non-existent attributes return null, we normalize to undefined
        return ret == null ?
            undefined :
            ret;
    }
},
removeAttr: function( elem, value ) {
    var name, propName,
        i = 0,
        attrNames = value && value.match( core_rnotwhite );

    if ( attrNames && elem.nodeType === 1 ) {
        while ( (name = attrNames[i++]) ) {
            propName = jQuery.propFix[ name ] || name;

            // Boolean attributes get special treatment (#10870)
            if ( jQuery.expr.match.bool.test( name ) ) {
                // Set corresponding property to false
                if ( getSetInput && getSetAttribute || !
ruseDefault.test( name ) ) {
                    elem[ propName ] = false;
                    // Support: IE<9
                    // Also clear defaultChecked/defaultSelected (if
appropriate)
                } else {
                    elem[ jQuery.camelCase( "default- " + name ) ] =
                    elem[ propName ] = false;
                }
            }
        }
    }
}

```

```

        // See #9699 for explanation of this approach (setting first,
then removal)
    } else {
        jQuery.attr( elem, name, "" );
    }

    elem.removeAttribute( getSetAttribute ? name : propName );
}
}

attrHooks: {
    type: {
        set: function( elem, value ) {
            if ( !jQuery.support.radioValue && value === "radio" &&
jQuery.nodeName(elem, "input") ) {
                // Setting the type on a radio button after the value
resets the value in IE6-9
                // Reset value to default in case type is set after value
during creation
                var val = elem.value;
                elem.setAttribute( "type", value );
                if ( val ) {
                    elem.value = val;
                }
                return value;
            }
        }
    }
},
};

propFix: {
    "for": "htmlFor",
    "class": "className"
},
};

prop: function( elem, name, value ) {
    var ret, hooks, notxml,
        nType = elem.nodeType;

    // don't get/set properties on text, comment and attribute nodes
    if ( !elem || nType === 3 || nType === 8 || nType === 2 ) {
        return;
    }
}

```

```

notxml = nType !== 1 || !jQuery.isXMLDoc( elem );

if ( notxml ) {
    // Fix name and attach hooks
    name = jQuery.propFix[ name ] || name;
    hooks = jQuery.propHooks[ name ];
}

if ( value !== undefined ) {
    return hooks && "set" in hooks && (ret = hooks.set( elem, value,
name )) !== undefined ?
        ret :
        ( elem[ name ] = value );
}

} else {
    return hooks && "get" in hooks && (ret = hooks.get( elem, name )) !
== null ?
        ret :
        elem[ name ];
},
};

propHooks: {
    tabIndex: {
        get: function( elem ) {
            // elem.tabIndex doesn't always return the correct value when
            it hasn't been explicitly set
            // http://fluidproject.org/blog/2008/01/09/getting-setting-and-
            removing-tabindex-values-with-javascript/
            // Use proper attribute retrieval(#12072)
            var tabindex = jQuery.find.attr( elem, "tabindex" );

            return tabindex ?
                parseInt( tabindex, 10 ) :
                rfocusable.test( elem.nodeName ) ||
rclickable.test( elem.nodeName ) && elem.href ?
                    0 :
                    -1;
            }
        }
    },
};

// Hooks for boolean attributes

```

```

boolHook = {
    set: function( elem, value, name ) {
        if ( value === false ) {
            // Remove boolean attributes when set to false
            jQueryremoveAttr( elem, name );
        } else if ( getSetInput && getSetAttribute || !ruseDefault.test( name ) ) {
            // IE<8 needs the *property* name
            elem.setAttribute( !getSetAttribute && jQuery.propFix[ name ] ||
name, name );
        }
        // Use defaultChecked and defaultSelected for oldIE
    } else {
        elem[ jQuery.camelCase( "default-" + name ) ] = elem[ name ] = true;
    }
    return name;
}
};

jQuery.each( jQuery.expr.match.bool.source.match( /\w+/g ), function( i, name ) {
    var getter = jQuery.expr.attrHandle[ name ] || jQuery.find.attr;

    jQuery.expr.attrHandle[ name ] = getSetInput && getSetAttribute || !
ruseDefault.test( name ) ?
        function( elem, name, isXML ) {
            var fn = jQuery.expr.attrHandle[ name ],
                ret = isXML ?
                    undefined :
                    /* jshint eqeqeq: false */
                    (jQuery.expr.attrHandle[ name ] = undefined) !=
getter( elem, name, isXML ) ?

                    name.toLowerCase() :
                    null;
            jQuery.expr.attrHandle[ name ] = fn;
            return ret;
        } :
        function( elem, name, isXML ) {
            return isXML ?
                undefined :
                elem[ jQuery.camelCase( "default-" + name ) ] ?
                    name.toLowerCase() :
                    null;
        };
});

```

```

// fix oldIE attributes
if ( !getSetInput || !getSetAttribute ) {
    jQuery.attrHooks.value = {
        set: function( elem, value, name ) {
            if ( jQuery.nodeName( elem, "input" ) ) {
                // Does not return so that setAttribute is also used
                elem.defaultValue = value;
            } else {
                // Use nodeHook if defined (#1954); otherwise setAttribute is
                fine
                return nodeHook && nodeHook.set( elem, value, name );
            }
        }
    };
}

// IE6/7 do not support getting/setting some attributes with get/setAttribute
if ( !getSetAttribute ) {

    // Use this for any attribute in IE6/7
    // This fixes almost every IE6/7 issue
    nodeHook = {
        set: function( elem, value, name ) {
            // Set the existing or create a new attribute node
            var ret = elem.getAttributeNode( name );
            if ( !ret ) {
                elem.setAttributeNode(
                    (ret = elem.ownerDocument.createAttribute( name )));
            }
        }

        ret.value = value += "";
    }

    // Break association with cloned elements by also using setAttribute
    (#9646)
    return name === "value" || value === elem.getAttribute( name ) ?
        value :
        undefined;
    }
};

jQuery.expr.attrHandle.id = jQuery.expr.attrHandle.name =
jQuery.expr.attrHandle.coords =
    // Some attributes are constructed with empty-string values when not
defined
    function( elem, name, isXML ) {

```

```

var ret;
return isXML ?
    undefined :
    (ret = elem.getAttributeNode( name )) && ret.value !== "" ?
        ret.value :
        null;
};

jQuery.valHooks.button = {
    get: function( elem, name ) {
        var ret = elem.getAttributeNode( name );
        return ret && ret.specified ?
            ret.value :
            undefined;
    },
    set: nodeHook.set
};

// Set contenteditable to false on removals(#10429)
// Setting to empty string throws an error as an invalid value
jQuery.attrHooks.contenteditable = {
    set: function( elem, value, name ) {
        nodeHook.set( elem, value === "" ? false : value, name );
    }
};

// Set width and height to auto instead of 0 on empty string( Bug #8150 )
// This is for removals
jQuery.each([ "width", "height" ], function( i, name ) {
    jQuery.attrHooks[ name ] = {
        set: function( elem, value ) {
            if ( value === "" ) {
                elem.setAttribute( name, "auto" );
                return value;
            }
        }
    };
});

// Some attributes require a special call on IE
// http://msdn.microsoft.com/en-us/library/ms536429%28VS.85%29.aspx
if ( !jQuery.support.hrefNormalized ) {
    // href/src property should get the full normalized URL (#10299/#12915)
    jQuery.each([ "href", "src" ], function( i, name ) {

```

```

jQuery.propHooks[ name ] = {
    get: function( elem ) {
        return elem.getAttribute( name, 4 );
    }
};

});

}

if ( !jQuery.support.style ) {
    jQuery.attrHooks.style = {
        get: function( elem ) {
            // Return undefined in the case of empty string
            // Note: IE uppercases css property names, but if we were
to .toLowerCase()
            // .cssText, that would destroy case sensitivity in URL's, like in
"background"
            return elem.style.cssText || undefined;
        },
        set: function( elem, value ) {
            return ( elem.style.cssText = value + " " );
        }
    };
}

// Safari mis-reports the default selected property of an option
// Accessing the parent's selectedIndex property fixes it
if ( !jQuery.support.optSelected ) {
    jQuery.propHooks.selected = {
        get: function( elem ) {
            var parent = elem.parentNode;

            if ( parent ) {
                parent.selectedIndex;

                // Make sure that it also works with optgroups, see #5701
                if ( parent.parentNode ) {
                    parent.parentNode.selectedIndex;
                }
            }
            return null;
        }
    };
}

jQuery.each([

```

```

    "tabIndex",
    "readOnly",
    "maxLength",
    "cellSpacing",
    "cellPadding",
    "rowSpan",
    "colSpan",
    "useMap",
    "frameBorder",
    "contentEditable"
], function() {
    jQuery.propFix[ this.toLowerCase() ] = this;
});

// IE6/7 call enctype encoding
if ( !jQuery.support.enctype ) {
    jQuery.propFix.enctype = "encoding";
}

// Radios and checkboxes getter/setter
jQuery.each([ "radio", "checkbox" ], function() {
    jQuery.valHooks[ this ] = {
        set: function( elem, value ) {
            if ( jQuery.isArray( value ) ) {
                return ( elem.checked = jQuery.inArray( jQuery(elem).val(),
value ) >= 0 );
            }
        }
    };
    if ( !jQuery.support.checkOn ) {
        jQuery.valHooks[ this ].get = function( elem ) {
            // Support: Webkit
            // "" is returned instead of "on" if a value isn't specified
            return elem.getAttribute("value") === null ? "on" : elem.value;
        };
    }
});
var rformElems = /^(?:input|select|textarea)$/i,
rkeyEvent = /^key/,
rmouseEvent = /^(?:mousecontextmenu|click)/,
rfocusMorph = /^(?:focusin|focus|focusout|blur)$/,
rtypenameSpace = /^([^.]*)(?:\.(.+))$/;

function returnTrue() {
    return true;
}

```

```

}

function returnFalse() {
    return false;
}

function safeActiveElement() {
    try {
        return document.activeElement;
    } catch ( err ) {}
}

/*
 * Helper functions for managing events -- not part of the public interface.
 * Props to Dean Edwards' addEvent library for many of the ideas.
 */
jQuery.event = {

    global: {},


    add: function( elem, types, handler, data, selector ) {
        var tmp, events, t, handleObjIn,
            special, eventHandle, handleObj,
            handlers, type, namespaces, origType,
            elemData = jQuery._data( elem );

        // Don't attach events to noData or text/comment nodes (but allow plain
        objects)
        if ( !elemData ) {
            return;
        }

        // Caller can pass in an object of custom data in lieu of the handler
        if ( handler.handler ) {
            handleObjIn = handler;
            handler = handleObjIn.handler;
            selector = handleObjIn.selector;
        }

        // Make sure that the handler has a unique ID, used to find/remove it later
        if ( !handler.guid ) {
            handler.guid = jQuery.guid++;
        }

        // Init the element's event structure and main handler, if this is the first

```

```

if ( !(events = elemData.events) ) {
    events = elemData.events = {};
}
if ( !(eventHandle = elemData.handle) ) {
    eventHandle = elemData.handle = function( e ) {
        // Discard the second event of a jQuery.event.trigger() and
        // when an event is called after a page has unloaded
        return typeof jQuery !== core_undefined && (!e ||
jQuery.event.triggered !== e.type) ?
            jQuery.event.dispatch.apply( eventHandle.elem,
arguments ) :
                undefined;
    };
    // Add elem as a property of the handle fn to prevent a memory leak
with IE non-native events
    eventHandle.elem = elem;
}

// Handle multiple events separated by a space
types = ( types || "" ).match( core_rnotwhite ) || [ "" ];
t = types.length;
while ( t-- ) {
    tmp = rtypenameexec( types[t] ) || [];
    type = origType = tmp[1];
    namespaces = ( tmp[2] || "" ).split( "." ).sort();

    // There *must* be a type, no attaching namespace-only handlers
    if ( !type ) {
        continue;
    }

    // If event changes its type, use the special event handlers for the
changed type
    special = jQuery.event.special[ type ] || {};

    // If selector defined, determine special event api type, otherwise
given type
    type = ( selector ? special.delegateType : special.bindType ) || type;

    // Update special based on newly reset type
    special = jQuery.event.special[ type ] || {};

    // handleObj is passed to all event handlers
    handleObj = jQuery.extend({
        type: type,

```

```

        origType: origType,
        data: data,
        handler: handler,
        guid: handler.guid,
        selector: selector,
        needsContext: selector &&
jQuery.expr.match.needsContext.test( selector ),
        namespace: namespaces.join(".")
}, handleObjIn );

// Init the event handler queue if we're the first
if ( !(handlers = events[ type ]) ) {
    handlers = events[ type ] = [];
    handlers.delegateCount = 0;

// Only use addEventListener/attachEvent if the special events
handler returns false
if ( !special.setup || special.setup.call( elem, data, namespaces,
eventHandle ) === false ) {
    // Bind the global event handler to the element
    if ( elem.addEventListener ) {
        elem.addEventListener( type, eventHandle,
false );

    } else if ( elem.attachEvent ) {
        elem.attachEvent( "on" + type, eventHandle );
    }
}

if ( special.add ) {
    special.add.call( elem, handleObj );

    if ( !handleObj.handler.guid ) {
        handleObj.handler.guid = handler.guid;
    }
}

// Add to the element's handler list, delegates in front
if ( selector ) {
    handlers.splice( handlers.delegateCount++, 0, handleObj );
} else {
    handlers.push( handleObj );
}

```

```

// Keep track of which events have ever been used, for event
optimization
    jQuery.event.global[ type ] = true;
}

// Nullify elem to prevent memory leaks in IE
elem = null;
},

// Detach an event or set of events from an element
remove: function( elem, types, handler, selector, mappedTypes ) {
    var j, handleObj, tmp,
        origCount, t, events,
        special, handlers, type,
        namespaces, origType,
        elemData = jQuery.hasData( elem ) && jQuery._data( elem );

    if ( !elemData || !(events = elemData.events) ) {
        return;
    }

    // Once for each type.namespace in types; type may be omitted
    types = ( types || "" ).match( core_rnotwhite ) || [""];
    t = types.length;
    while ( t-- ) {
        tmp = rtypenamespace.exec( types[t] ) || [];
        type = origType = tmp[1];
        namespaces = ( tmp[2] || "" ).split( "." ).sort();

        // Unbind all events (on this namespace, if provided) for the element
        if ( !type ) {
            for ( type in events ) {
                jQuery.event.remove( elem, type + types[ t ], handler,
selector, true );
            }
            continue;
        }

        special = jQuery.event.special[ type ] || {};
        type = ( selector ? special.delegateType : special.bindType ) || type;
        handlers = events[ type ] || [];
        tmp = tmp[2] && new RegExp( "(^|\\.)" + namespaces.join("\\".(?:.*\ \\
\.)") + "(\\\\.I$)" );

```

// Remove matching events

```

origCount = j = handlers.length;
while ( j-- ) {
    handleObj = handlers[ j ];

    if ( ( mappedTypes || origType === handleObj.origType ) &&
        ( !handler || handler.guid === handleObj.guid ) &&
        ( !tmp || tmp.test( handleObj.namespace ) ) &&
        ( !selector || selector === handleObj.selector || selector
===== "*" && handleObj.selector ) ) {
        handlers.splice( j, 1 );

        if ( handleObj.selector ) {
            handlers.delegateCount--;
        }
        if ( special.remove ) {
            special.remove.call( elem, handleObj );
        }
    }
}

// Remove generic event handler if we removed something and no
more handlers exist
// (avoids potential for endless recursion during removal of special
event handlers)
if ( origCount && !handlers.length ) {
    if ( !special.teardown || special.teardown.call( elem,
namespaces, elemData.handle ) === false ) {
        jQuery.removeEvent( elem, type, elemData.handle );
    }

    delete events[ type ];
}
}

// Remove the expando if it's no longer used
if ( jQuery.isEmptyObject( events ) ) {
    delete elemData.handle;

    // removeData also checks for emptiness and clears the expando if
empty
    // so use it instead of delete
    jQuery._removeData( elem, "events" );
}
},

```

```

trigger: function( event, data, elem, onlyHandlers ) {
    var handle, ontype, cur,
        bubbleType, special, tmp, i,
        eventPath = [ elem || document ],
        type = core_hasOwn.call( event, "type" ) ? event.type : event,
        namespaces = core_hasOwn.call( event, "namespace" ) ?
event.namespace.split( "." ) : [];

    cur = tmp = elem = elem || document;

    // Don't do events on text and comment nodes
    if ( elem.nodeType === 3 || elem.nodeType === 8 ) {
        return;
    }

    // focus/blur morphs to focusin/out; ensure we're not firing them right now
    if ( rfocusMorph.test( type + jQuery.event.triggered ) ) {
        return;
    }

    if ( type.indexOf( "." ) >= 0 ) {
        // Namespaced trigger; create a regexp to match event type in
handle()
        namespaces = type.split( "." );
        type = namespaces.shift();
        namespaces.sort();
    }
    ontype = type.indexOf( ":" ) < 0 && "on" + type;

    // Caller can pass in a jQuery.Event object, Object, or just an event type
string
    event = event[ jQuery.expando ] ?
        event :
        new jQuery.Event( type, typeof event === "object" && event );

    // Trigger bitmask: & 1 for native handlers; & 2 for jQuery (always true)
    event.isTrigger = onlyHandlers ? 2 : 3;
    event.namespace = namespaces.join( "." );
    event.namespace_re = event.namespace ?
        new RegExp( "(^|\\.)" + namespaces.join( "\\.(?:.*\\.|)" ) + "(\\.|$)" ) :
        null;

    // Clean up the event in case it is being reused
    event.result = undefined;
    if ( !event.target ) {

```

```

        event.target = elem;
    }

    // Clone any incoming data and prepend the event, creating the handler arg
list
    data = data == null ?
        [ event ]:
    jQuery.makeArray( data, [ event ] );

    // Allow special events to draw outside the lines
    special = jQuery.event.special[ type ] || {};
    if ( !onlyHandlers && special.trigger && special.trigger.apply( elem, data )
==== false ) {
        return;
    }

    // Determine event propagation path in advance, per W3C events spec
(#9951)
    // Bubble up to document, then to window; watch for a global
ownerDocument var (#9724)
    if ( !onlyHandlers && !special.noBubble && !jQuery.isWindow( elem ) ) {

        bubbleType = special.delegateType || type;
        if ( !rfocusMorph.test( bubbleType + type ) ) {
            cur = cur.parentNode;
        }
        for ( ; cur; cur = cur.parentNode ) {
            eventPath.push( cur );
            tmp = cur;
        }

        // Only add window if we got to document (e.g., not plain obj or
detached DOM)
        if ( tmp === (elem.ownerDocument || document) ) {
            eventPath.push( tmp.defaultView || tmp.parentWindow ||
window );
        }
    }

    // Fire handlers on the event path
    i = 0;
    while ( (cur = eventPath[i++]) && !event.isPropagationStopped() ) {

        event.type = i > 1 ?
            bubbleType :

```

```

special.bindType || type;

// jQuery handler
handle = (jQuery._data( cur, "events" ) || {})[ event.type ] &&
jQuery._data( cur, "handle" );
if ( handle ) {
    handle.apply( cur, data );
}

// Native handler
handle = ontype && cur[ ontype ];
if ( handle && jQuery.acceptData( cur ) && handle.apply &&
handle.apply( cur, data ) === false ) {
    event.preventDefault();
}
event.type = type;

// If nobody prevented the default action, do it now
if ( !onlyHandlers && !event.isDefaultPrevented() ) {

    if ( (!special._default || special._default.apply( eventPath.pop(), data ) )
==== false) &&
        jQuery.acceptData( elem ) ) {

            // Call a native DOM method on the target with the same
name name as the event.
            // Can't use an .isFunction() check here because IE6/7 fails that
test.
            // Don't do default actions on window, that's where global
variables be (#6170)
            if ( ontype && elem[ type ] && !jQuery.isWindow( elem ) ) {

                // Don't re-trigger an onFOO event when we call its
FOO() method
                tmp = elem[ ontype ];

                if ( tmp ) {
                    elem[ ontype ] = null;
                }

                // Prevent re-triggering of the same event, since we
already bubbled it above
                jQuery.event.triggered = type;
                try {

```

```

        elem[ type ]0;
    } catch ( e ) {
        // IE<9 dies on focus/blur to hidden element
(#1486,#12518)                                // only reproducible on winXP IE8 native, not IE9
in IE8 mode
    }
    jQuery.event.triggered = undefined;

    if ( tmp ) {
        elem[ ontype ] = tmp;
    }
}
}

return event.result;
},

dispatch: function( event ) {

// Make a writable jQuery.Event from the native event object
event = jQuery.event.fix( event );

var i, ret, handleObj, matched, j,
    handlerQueue = [],
    args = core_slice.call( arguments ),
    handlers = ( jQuery._data( this, "events" ) || {} )[ event.type ] || [],
    special = jQuery.event.special[ event.type ] || {};

// Use the fix-ed jQuery.Event rather than the (read-only) native event
args[0] = event;
event.delegateTarget = this;

// Call the preDispatch hook for the mapped type, and let it bail if desired
if ( special.preDispatch && special.preDispatch.call( this, event ) === false ) {
    return;
}

// Determine handlers
handlerQueue = jQuery.event.handlers.call( this, event, handlers );

// Run delegates first; they may want to stop propagation beneath us
i = 0;

```

```

        while ( (matched = handlerQueue[ i++ ]) && !event.isPropagationStopped() )
{
    event.currentTarget = matched.elem;

    j = 0;
    while ( (handleObj = matched.handlers[ j++ ]) && !
event.isImmediatePropagationStopped() ) {

        // Triggered event must either 1) have no namespace, or
        // 2) have namespace(s) a subset or equal to those in the
bound event (both can have no namespace).
        if ( !event.namespace_re ||
event.namespace_re.test( handleObj.namespace ) ) {

            event.handleObj = handleObj;
            event.data = handleObj.data;

            ret = ( (jQuery.event.special[ handleObj.origType ] ||
{}).handle || handleObj.handler )
                .apply( matched.elem, args );

            if ( ret !== undefined ) {
                if ( (event.result = ret) === false ) {
                    event.preventDefault();
                    event.stopPropagation();
                }
            }
        }
    }

    // Call the postDispatch hook for the mapped type
    if ( special.postDispatch ) {
        special.postDispatch.call( this, event );
    }

    return event.result;
},

```

handlers: function(event, handlers) {

```

    var sel, handleObj, matches, i,
        handlerQueue = [],
        delegateCount = handlers.delegateCount,
        cur = event.target;

```

```

// Find delegate handlers
// Black-hole SVG <use> instance trees (#13180)
// Avoid non-left-click bubbling in Firefox (#3861)
if ( delegateCount && cur.nodeType && (!event.button || event.type !==
"click") ) {

    /* jshint eqeqeq: false */
    for ( ; cur != this; cur = cur.parentNode || this ) {
        /* jshint eqeqeq: true */

        // Don't check non-elements (#13208)
        // Don't process clicks on disabled elements (#6911, #8165,
#11382, #11764)
        if ( cur.nodeType === 1 && (cur.disabled !== true ||
event.type !== "click") ) {
            matches = [];
            for ( i = 0; i < delegateCount; i++ ) {
                handleObj = handlers[ i ];

                // Don't conflict with Object.prototype properties
(#13203)
                sel = handleObj.selector + " ";

                if ( matches[ sel ] === undefined ) {
                    matches[ sel ] =
handleObj.needsContext ?
                        jQuery( sel, this ).index( cur ) >= 0 :
                        jQuery.find( sel, this, null,
[ cur ].length;
                }
                if ( matches[ sel ] ) {
                    matches.push( handleObj );
                }
            }
            if ( matches.length ) {
                handlerQueue.push({ elem: cur, handlers:
matches });
            }
        }
    }

    // Add the remaining (directly-bound) handlers
    if ( delegateCount < handlers.length ) {

```

```

        handlerQueue.push({ elem: this, handlers:
handlers.slice( delegateCount ) });
    }

    return handlerQueue;
},

fix: function( event ) {
    if ( event[ jQuery.expando ] ) {
        return event;
    }

// Create a writable copy of the event object and normalize some properties
var i, prop, copy,
    type = event.type,
    originalEvent = event,
    fixHook = this.fixHooks[ type ];

if ( !fixHook ) {
    this.fixHooks[ type ] = fixHook =
        rmouseEvent.test( type ) ? this.mouseHooks :
        rkeyEvent.test( type ) ? this.keyHooks :
        {};
}
copy = fixHook.props ? this.props.concat( fixHook.props ) : this.props;

event = new jQuery.Event( originalEvent );

i = copy.length;
while ( i-- ) {
    prop = copy[ i ];
    event[ prop ] = originalEvent[ prop ];
}

// Support: IE<9
// Fix target property (#1925)
if ( !event.target ) {
    event.target = originalEvent.srcElement || document;
}

// Support: Chrome 23+, Safari?
// Target should not be a text node (#504, #13143)
if ( event.target.nodeType === 3 ) {
    event.target = event.target.parentNode;
}

```

```

// Support: IE<9
// For mouse/key events, metaKey==false if it's undefined (#3368, #11328)
event.metaKey = !!event.metaKey;

return fixHook.filter ? fixHook.filter( event, originalEvent ) : event;
},

// Includes some event props shared by KeyEvent and MouseEvent
props: "altKey bubbles cancelable ctrlKey currentTarget eventPhase metaKey
relatedTarget shiftKey target timeStamp view which".split(" "),

fixHooks: {},


keyHooks: {
    props: "char charCode key keyCode".split(" "),
    filter: function( event, original ) {

        // Add which for key events
        if ( event.which == null ) {
            event.which = original.charCode != null ? original.charCode :
original.keyCode;
        }

        return event;
    }
},
mouseHooks: {
    props: "button buttons clientX clientY fromElement offsetX offsetY pageX
pageY screenX screenY toElement".split(" "),
    filter: function( event, original ) {
        var body, eventDoc, doc,
            button = original.button,
            fromElement = original.fromElement;

        // Calculate pageX/Y if missing and clientX/Y available
        if ( event.pageX == null && original.clientX != null ) {
            eventDoc = event.target.ownerDocument || document;
            doc = eventDoc.documentElement;
            body = eventDoc.body;

            event.pageX = original.clientX + ( doc && doc.scrollLeft ||
body && body.scrollLeft || 0 ) - ( doc && doc.clientLeft || body && body.clientLeft || 0 );
        }
    }
},

```

```

        event.pageY = original.clientY + ( doc && doc.scrollTop || 
body && body.scrollTop || 0 ) - ( doc && doc.clientTop || body && body.clientTop || 0 );
    }

    // Add relatedTarget, if necessary
    if ( !event.relatedTarget && fromElement ) {
        event.relatedTarget = fromElement === event.target ?
original.toElement : fromElement;
    }

    // Add which for click: 1 === left; 2 === middle; 3 === right
    // Note: button is not normalized, so don't use it
    if ( !event.which && button !== undefined ) {
        event.which = ( button & 1 ? 1 : ( button & 2 ? 3 : ( button & 4 ?
2 : 0 ) ) );
    }

    return event;
},
special: {
    load: {
        // Prevent triggered image.load events from bubbling to window.load
        noBubble: true
    },
    focus: {
        // Fire native event if possible so blur/focus sequence is correct
        trigger: function() {
            if ( this !== safeActiveElement() && this.focus ) {
                try {
                    this.focus();
                    return false;
                } catch ( e ) {
                    // Support: IE<9
                    // If we error on focus to hidden element (#1486,
#12518),
                    // let .trigger() run the handlers
                }
            }
        },
        delegateType: "focusin"
    },
    blur: {
        trigger: function() {

```

```

        if ( this === safeActiveElement() && this.blur ) {
            this.blur();
            return false;
        }
    },
    delegateType: "focusout"
},
click: {
    // For checkbox, fire native event so checked state will be right
    trigger: function() {
        if ( jQuery.nodeName( this, "input" ) && this.type ===
"checkbox" && this.click ) {
            this.click();
            return false;
        }
    },
    // For cross-browser consistency, don't fire native .click() on links
    _default: function( event ) {
        return jQuery.nodeName( event.target, "a" );
    }
},
beforeunload: {
    postDispatch: function( event ) {

        // Even when returnValue equals to undefined Firefox will still
show alert
        if ( event.result !== undefined ) {
            event.originalEvent.returnValue = event.result;
        }
    }
},
simulate: function( type, elem, event, bubble ) {
    // Piggyback on a donor event to simulate a different one.
    // Fake originalEvent to avoid donor's stopPropagation, but if the
    // simulated event prevents default then we do the same on the donor.
    var e = jQuery.extend(
        new jQuery.Event(),
        event,
        {
            type: type,
            isSimulated: true,

```

```

        originalEvent: {}
    }
);
if ( bubble ) {
    jQuery.event.trigger( e, null, elem );
} else {
    jQuery.event.dispatch.call( elem, e );
}
if ( e.isDefaultPrevented() ) {
    event.preventDefault();
}
}

};

jQuery.removeEventListener = document.removeEventListener ?
function( elem, type, handle ) {
    if ( elem.removeEventListener ) {
        elem.removeEventListener( type, handle, false );
    }
} :
function( elem, type, handle ) {
    var name = "on" + type;

    if ( elem.detachEvent ) {

        // #8545, #7054, preventing memory leaks for custom events in IE6-8
        // detachEvent needed property on element, by name of that event,
        to properly expose it to GC
        if ( typeof elem[ name ] === core_undefined ) {
            elem[ name ] = null;
        }

        elem.detachEvent( name, handle );
    }
};

jQuery.Event = function( src, props ) {
// Allow instantiation without the 'new' keyword
if ( !(this instanceof jQuery.Event) ) {
    return new jQuery.Event( src, props );
}

// Event object
if ( src && src.type ) {
    this.originalEvent = src;
}

```

```

this.type = src.type;

// Events bubbling up the document may have been marked as prevented
// by a handler lower down the tree; reflect the correct value.
this.isDefaultPrevented = ( src.defaultPrevented || src.returnValue === false ||

    src.getPreventDefault && src.getPreventDefault() ) ? returnTrue :

returnFalse;

// Event type
} else {
    this.type = src;
}

// Put explicitly provided properties onto the event object
if ( props ) {
    jQuery.extend( this, props );
}

// Create a timestamp if incoming event doesn't have one
this.timeStamp = src && src.timeStamp || jQuery.now();

// Mark it as fixed
this[ jQuery.expando ] = true;
};

// jQuery.Event is based on DOM3 Events as specified by the ECMAScript Language
Binding
// http://www.w3.org/TR/2003/WD-DOM-Level-3-Events-20030331/ecma-script-
binding.html
jQuery.Event.prototype = {
    isDefaultPrevented: returnFalse,
    isPropagationStopped: returnFalse,
    isImmediatePropagationStopped: returnFalse,

    preventDefault: function() {
        var e = this.originalEvent;

        this.isDefaultPrevented = returnTrue;
        if ( !e ) {
            return;
        }

        // If preventDefault exists, run it on the original event
        if ( e.preventDefault ) {
            e.preventDefault();
        }
    }
}

```

```

// Support: IE
// Otherwise set the returnValue property of the original event to false
} else {
    e.returnValue = false;
}
},
stopPropagation: function() {
    var e = this.originalEvent;

    this.isPropagationStopped = returnTrue;
    if ( !e ) {
        return;
    }
    // If stopPropagation exists, run it on the original event
    if ( e.stopPropagation ) {
        e.stopPropagation();
    }

    // Support: IE
    // Set the cancelBubble property of the original event to true
    e.cancelBubble = true;
},
stopImmediatePropagation: function() {
    this.isImmediatePropagationStopped = returnTrue;
    this.stopPropagation();
}
};

// Create mouseenter/leave events using mouseover/out and event-time checks
jQuery.each({
    mouseenter: "mouseover",
   mouseleave: "mouseout"
}, function( orig, fix ) {
    jQuery.event.special[ orig ] = {
        delegateType: fix,
        bindType: fix,

        handle: function( event ) {
            var ret,
                target = this,
                related = event.relatedTarget,
                handleObj = event.handleObj;

            // For mousenter/leave call the handler if related is outside the target.
        }
    };
});

```

```

// NB: No relatedTarget if the mouse left/entered the browser
window
    if ( !related || (related !== target && !jQuery.contains( target,
related )) ) {
        event.type = handleObj.origType;
        ret = handleObj.handler.apply( this, arguments );
        event.type = fix;
    }
    return ret;
}
};

// IE submit delegation
if ( !jQuery.support.submitBubbles ) {

    jQuery.event.special.submit = {
        setup: function() {
            // Only need this for delegated form submit events
            if ( jQuery.nodeName( this, "form" ) ) {
                return false;
            }

            // Lazy-add a submit handler when a descendant form may
            potentially be submitted
            jQuery.event.add( this, "click._submit keypress._submit", function( e )
{
            // Node name check avoids a VML-related crash in IE (#9807)
            var elem = e.target,
                form = jQuery.nodeName( elem, "input" ) ||
jQuery.nodeName( elem, "button" ) ? elem.form : undefined;
            if ( form && !jQuery._data( form, "submitBubbles" ) ) {
                jQuery.event.add( form, "submit._submit",
function( event ) {
                event._submit_bubble = true;
            });
                jQuery._data( form, "submitBubbles", true );
            }
        });
        // return undefined since we don't need an event listener
    },
    postDispatch: function( event ) {
        // If form was submitted by the user, bubble the event up the tree
        if ( event._submit_bubble ) {

```

```

        delete event._submit_bubble;
        if ( this.parentNode && !event.isTrigger ) {
            jQuery.event.simulate( "submit", this.parentNode,
event, true );
        }
    }
},
};

teardown: function() {
    // Only need this for delegated form submit events
    if ( jQuery.nodeName( this, "form" ) ) {
        return false;
    }

    // Remove delegated handlers; cleanData eventually reaps submit
    // handlers attached above
    jQuery.event.remove( this, "._submit" );
}
};

// IE change delegation and checkbox/radio fix
if ( !jQuery.support.changeBubbles ){

    jQuery.event.special.change = {

        setup: function() {

            if ( rformElems.test( this.nodeName ) ) {
                // IE doesn't fire change on a check/radio until blur; trigger it
                // on click
                // after a propertychange. Eat the blur-change in
                // special.change.handle.
                // This still fires onchange a second time for check/radio after
                // blur.
                if ( this.type === "checkbox" || this.type === "radio" ) {
                    jQuery.event.add( this, "propertychange._change",
function( event ) {
                        if ( event.originalEvent.propertyName ===
                            "checked" ) {
                            this._just_changed = true;
                        }
                    });
                    jQuery.event.add( this, "click._change", function( event )
{
}
```
```

```

 if (this._just_changed && !event.isTrigger) {
 this._just_changed = false;
 }
 // Allow triggered, simulated change events
(#11500) jQuery.event.simulate("change", this, event, true
);
 });
 }
 return false;
}
// Delegated event; lazy-add a change handler on descendant inputs
jQuery.event.add(this, "beforeactivate._change", function(e) {
 var elem = e.target;

 if (rformElems.test(elem.nodeName) && !jQuery._data(elem,
"changeBubbles")){
 jQuery.event.add(elem, "change._change",
function(event) {
 if (this.parentNode && !event.isSimulated && !
event.isTrigger) {
 jQuery.event.simulate("change",
this.parentNode, event, true);
 }
 });
 jQuery._data(elem, "changeBubbles", true);
 }
});
},
handle: function(event) {
 var elem = event.target;

 // Swallow native change events from checkbox/radio, we already
triggered them above
 if (this !== elem || event.isSimulated || event.isTrigger || (elem.type !=
"radio" && elem.type != "checkbox")){
 return event.handleObj.handler.apply(this, arguments);
 }
},
teardown: function() {
 jQuery.event.remove(this, ".change");

 return !rformElems.test(this.nodeName);
}

```

```

 }
 };

}

// Create "bubbling" focus and blur events
if (!jQuery.support.focusinBubbles) {
 jQuery.each({ focus: "focusin", blur: "focusout" }, function(orig, fix) {

 // Attach a single capturing handler while someone wants focusin/focusout
 var attaches = 0,
 handler = function(event) {
 jQuery.event.simulate(fix, event.target,
 jQuery.event.fix(event), true);
 };

 jQuery.event.special[fix] = {
 setup: function() {
 if (attaches++ === 0) {
 document.addEventListener(orig, handler, true);
 }
 },
 teardown: function() {
 if (--attaches === 0) {
 document.removeEventListener(orig, handler, true);
 }
 }
 };
 });
}

jQuery.fn.extend({
 on: function(types, selector, data, fn, /*INTERNAL*/ one) {
 var type, origFn;

 // Types can be a map of types/handlers
 if (typeof types === "object") {
 // (types-Object, selector, data)
 if (typeof selector !== "string") {
 // (types-Object, data)
 data = data || selector;
 selector = undefined;
 }
 for (type in types) {
 this.on(type, selector, data, types[type], one);
 }
 }
 }
});

```

```

 }
 return this;
 }

 if (data == null && fn == null) {
 // (types, fn)
 fn = selector;
 data = selector = undefined;
 } else if (fn == null) {
 if (typeof selector === "string") {
 // (types, selector, fn)
 fn = data;
 data = undefined;
 } else {
 // (types, data, fn)
 fn = data;
 data = selector;
 selector = undefined;
 }
 }
 if (fn === false) {
 fn = returnFalse;
 } else if (!fn) {
 return this;
 }

 if (one === 1) {
 origFn = fn;
 fn = function(event) {
 // Can use an empty set, since event contains the info
 jQuery().off(event);
 return origFn.apply(this, arguments);
 };
 // Use same guid so caller can remove using origFn
 fn.guid = origFn.guid || (origFn.guid = jQuery.guid++);
 }
 return this.each(function() {
 jQuery.event.add(this, types, fn, data, selector);
 });
},
one: function(types, selector, data, fn) {
 return this.on(types, selector, data, fn, 1);
},
off: function(types, selector, fn) {
 var handleObj, type;
}

```

```

if (types && types.preventDefault && types.handleObj) {
 // (event) dispatched jQuery.Event
 handleObj = types.handleObj;
 jQuery(types.delegateTarget).off(
 handleObj.namespace ? handleObj.origType + "." +
handleObj.namespace : handleObj.origType,
 handleObj.selector,
 handleObj.handler
);
 return this;
}
if (typeof types === "object") {
 // (types-object [, selector])
 for (type in types) {
 this.off(type, selector, types[type]);
 }
 return this;
}
if (selector === false || typeof selector === "function") {
 // (types [, fn])
 fn = selector;
 selector = undefined;
}
if (fn === false) {
 fn = returnFalse;
}
return this.each(function() {
 jQuery.event.remove(this, types, fn, selector);
});
},
trigger: function(type, data) {
 return this.each(function() {
 jQuery.event.trigger(type, data, this);
 });
},
triggerHandler: function(type, data) {
 var elem = this[0];
 if (elem) {
 return jQuery.event.trigger(type, data, elem, true);
 }
}
});
var isSimple = /^#[\.\w]*$/,
rparentsprev = /^(?:parents|prev(?:Until|All))$/,

```

```

rneedsContext = jQuery.expr.match.needsContext,
// methods guaranteed to produce a unique set when starting from a unique set
guaranteedUnique = {
 children: true,
 contents: true,
 next: true,
 prev: true
};

jQuery.fn.extend({
 find: function(selector) {
 var i,
 ret = [],
 self = this,
 len = self.length;

 if (typeof selector !== "string") {
 return this.pushStack(jQuery(selector).filter(function() {
 for (i = 0; i < len; i++) {
 if (jQuery.contains(self[i], this)) {
 return true;
 }
 }
 }));
 }

 for (i = 0; i < len; i++) {
 jQuery.find(selector, self[i], ret);
 }

 // Needed because $(selector, context) becomes $(context).find(selector)
 ret = this.pushStack(len > 1 ? jQuery.unique(ret) : ret);
 ret.selector = this.selector ? this.selector + " " + selector : selector;
 return ret;
 },
 has: function(target) {
 var i,
 targets = jQuery(target, this),
 len = targets.length;

 return this.filter(function() {
 for (i = 0; i < len; i++) {
 if (jQuery.contains(this, targets[i])) {
 return true;
 }
 }
 });
 }
});

```

```

 }
 }
};

},
not: function(selector) {
 return this.pushStack(winnow(this, selector || [], true));
},
filter: function(selector) {
 return this.pushStack(winnow(this, selector || [], false));
},
is: function(selector) {
 return !!winnow(
 this,
 // If this is a positional/relative selector, check membership in the
 // returned set
 // so $("p:first").is("p:last") won't return true for a doc with two "p".
 typeof selector === "string" && rneedsContext.test(selector) ?
 jQuery(selector) :
 selector || [],
 false
).length;
},
closest: function(selectors, context) {
 var cur,
 i = 0,
 l = this.length,
 ret = [],
 pos = rneedsContext.test(selectors) || typeof selectors !== "string" ?
 jQuery(selectors, context || this.context) :
 0;

 for (; i < l; i++) {
 for (cur = this[i]; cur && cur !== context; cur = cur.parentNode) {
 // Always skip document fragments
 if (cur.nodeType < 11 && (pos ?
 pos.index(cur) > -1 :

 // Don't pass non-elements to Sizzle
 cur.nodeType === 1 &&
 jQuery.find.matchesSelector(cur, selectors))) {

```

```

 cur = ret.push(cur);
 break;
 }
}

return this.pushStack(ret.length > 1 ? jQuery.unique(ret) : ret);
},

// Determine the position of an element within
// the matched set of elements
index: function(elem) {

 // No argument, return index in parent
 if (!elem) {
 return (this[0] && this[0].parentNode) ? this.first().prevAll().length : -1;
 }

 // index in selector
 if (typeof elem === "string") {
 return jQuery.inArray(this[0], jQuery(elem));
 }

 // Locate the position of the desired element
 return jQuery.inArray(
 // If it receives a jQuery object, the first element is used
 elem.jquery ? elem[0] : elem, this);
},

add: function(selector, context) {
 var set = typeof selector === "string" ?
 jQuery(selector, context) :
 jQuery.makeArray(selector && selector.nodeType ?
 [selector] : selector),
 all = jQuery.merge(this.get(), set);

 return this.pushStack(jQuery.unique(all));
},

addBack: function(selector) {
 return this.add(selector == null ?
 this.prevObject : this.prevObject.filter(selector)
);
}

```

```

});
```

```

function sibling(cur, dir) {
 do {
 cur = cur[dir];
 } while (cur && cur.nodeType !== 1);
```

```

 return cur;
}
```

```

jQuery.each({
 parent: function(elem) {
 var parent = elem.parentNode;
 return parent && parent.nodeType !== 11 ? parent : null;
 },
 parents: function(elem) {
 return jQuery.dir(elem, "parentNode");
 },
 parentsUntil: function(elem, i, until) {
 return jQuery.dir(elem, "parentNode", until);
 },
 next: function(elem) {
 return sibling(elem, "nextSibling");
 },
 prev: function(elem) {
 return sibling(elem, "previousSibling");
 },
 nextAll: function(elem) {
 return jQuery.dir(elem, "nextSibling");
 },
 prevAll: function(elem) {
 return jQuery.dir(elem, "previousSibling");
 },
 nextUntil: function(elem, i, until) {
 return jQuery.dir(elem, "nextSibling", until);
 },
 prevUntil: function(elem, i, until) {
 return jQuery.dir(elem, "previousSibling", until);
 },
 siblings: function(elem) {
 return jQuery.sibling((elem.parentNode || {}).firstChild, elem);
 },
 children: function(elem) {
 return jQuery.sibling(elem.firstChild);
 },

```

```

contents: function(elem) {
 return jQuery.nodeName(elem, "iframe") ?
 elem.contentDocument || elem.contentWindow.document :
 jQuery.merge([], elem.childNodes);
}
}, function(name, fn) {
 jQuery.fn[name] = function(until, selector) {
 var ret = jQuery.map(this, fn, until);

 if (name.slice(-5) !== "Until") {
 selector = until;
 }

 if (selector && typeof selector === "string") {
 ret = jQuery.filter(selector, ret);
 }

 if (this.length > 1) {
 // Remove duplicates
 if (!guaranteedUnique[name]) {
 ret = jQuery.unique(ret);
 }

 // Reverse order for parents* and prev-derivatives
 if (rparentsprev.test(name)) {
 ret = ret.reverse();
 }
 }
 }

 return this.pushStack(ret);
};

jQuery.extend({
 filter: function(expr, elems, not) {
 var elem = elems[0];

 if (not) {
 expr = ":not(" + expr + ")";
 }

 return elems.length === 1 && elem.nodeType === 1 ?
 jQuery.find.matchesSelector(elem, expr) ? [elem] : [] :
 jQuery.find.matches(expr, jQuery.grep(elems, function(elem) {
 return elem.nodeType === 1;
 })
);
 }
});

```

```

 });
 },

 dir: function(elem, dir, until) {
 var matched = [],
 cur = elem[dir];

 while (cur && cur.nodeType !== 9 && (until === undefined || cur.nodeType !
== 1 || !jQuery(cur).is(until))) {
 if (cur.nodeType === 1) {
 matched.push(cur);
 }
 cur = cur[dir];
 }
 return matched;
 },
 sibling: function(n, elem) {
 var r = [];

 for (; n; n = n.nextSibling) {
 if (n.nodeType === 1 && n !== elem) {
 r.push(n);
 }
 }

 return r;
 }
});

// Implement the identical functionality for filter and not
function winnow(elements, qualifier, not) {
 if (jQuery.isFunction(qualifier)) {
 return jQuery.grep(elements, function(elem, i) {
 /* jshint -W018 */
 return !!qualifier.call(elem, i, elem) !== not;
 });
 }

 if (qualifier.nodeType) {
 return jQuery.grep(elements, function(elem) {
 return (elem === qualifier) !== not;
 });
 }
}

```

```

}

if (typeof qualifier === "string") {
 if (isSimple.test(qualifier)) {
 return jQuery.filter(qualifier, elements, not);
 }

 qualifier = jQuery.filter(qualifier, elements);
}

return jQuery.grep(elements, function(elem) {
 return (jQuery.inArray(elem, qualifier) >= 0) !== not;
});
}

function createSafeFragment(document) {
 var list = nodeNames.split(" "),
 safeFrag = document.createDocumentFragment();

 if (safeFrag.createElement) {
 while (list.length) {
 safeFrag.createElement(
 list.pop()
);
 }
 }
 return safeFrag;
}

var nodeNames = "abbr|article|aside|audio|bd|canvas|datalist|details|figcaption|"
 + "figure|footer|"
 + "header|hgroup|mark|meter|nav|output|progress|section|summary|time|"
 + "video",
 rinlinejQuery = / jQuery\d+="(?:null|\d+)"/g,
 rnoshimcache = new RegExp("<(?:" + nodeNames + ")[\\s/>]", "i"),
 rleadingWhitespace = /^\\s+/,
 rxhtmlITag = /<(?![^>]+real|col|embed|hr|img|input|link|meta|param)([\\w:]+)[^>]*\\/>/gi,
 rtagName = /<([\\w:]+)/,
 rtbody = /<tbody/i,
 rhtml = /<|&#?\\w+;/,
 rnolnnerhtml = /<(?:(script|style|link)/i,
 manipulation_rcheckableType = /(^|(?:(checkbox|radio)$)/i,
 // checked="checked" or checked
 rchecked = /checked\\s*(?:[^=]|=\\s*.checked.)/i,
 rscriptType = /\\$|^(?:(java|ecma)script)/i,

```

```

rscriptTypeMasked = /^true\/(.*)/,
rcleanScript = /\s*<!(?:\[CDATA\[|\-\-|(?:\]\])|-->\s*/g,

// We have to close these tags to support XHTML (#13200)
wrapMap = {
 option: [1, "<select multiple='multiple'>", "</select>"],
 legend: [1, "<fieldset>", "</fieldset>"],
 area: [1, "<map>", "</map>"],
 param: [1, "<object>", "</object>"],
 thead: [1, "<table>", "</table>"],
 tr: [2, "<table><tbody>", "</tbody></table>"],
 col: [2, "<table><tbody></tbody><colgroup>", "</colgroup></table>"],
 td: [3, "<table><tbody><tr>", "</tr></tbody></table>"],

 // IE6-8 can't serialize link, script, style, or any html5 (NoScope) tags,
 // unless wrapped in a div with non-breaking characters in front of it.
 _default: jQuery.support.htmlSerialize ? [0, "", ""] : [1, "X<div>", "</div>"]
},
safeFragment = createSafeFragment(document),
fragmentDiv = safeFragment.appendChild(document.createElement("div"));

wrapMap.optgroup = wrapMap.option;
wrapMap.tbody = wrapMap.tfoot = wrapMap.colgroup = wrapMap.caption =
wrapMap.thead;
wrapMap.th = wrapMap.td;

jQuery.fn.extend({
 text: function(value) {
 return jQuery.access(this, function(value) {
 return value === undefined ?
 jQuery.text(this) :
 this.empty().append((this[0] && this[0].ownerDocument ||
document).createTextNode(value));
 }, null, value, arguments.length);
 },
 append: function() {
 return this.domManip(arguments, function(elem) {
 if (this.nodeType === 1 || this.nodeType === 11 || this.nodeType
=== 9) {
 var target = manipulationTarget(this, elem);
 target.appendChild(elem);
 }
 });
 }
});

```

```

 },
 prepend: function() {
 return this.domManip(arguments, function(elem) {
 if (this.nodeType === 1 || this.nodeType === 11 || this.nodeType
==== 9) {
 var target = manipulationTarget(this, elem);
 target.insertBefore(elem, target.firstChild);
 }
 });
 },
 before: function() {
 return this.domManip(arguments, function(elem) {
 if (this.parentNode) {
 this.parentNode.insertBefore(elem, this);
 }
 });
 },
 after: function() {
 return this.domManip(arguments, function(elem) {
 if (this.parentNode) {
 this.parentNode.insertBefore(elem, this.nextSibling);
 }
 });
 },
 // keepData is for internal use only--do not document
 remove: function(selector, keepData) {
 var elem,
 elems = selector ? jQuery.filter(selector, this) : this,
 i = 0;

 for (; (elem = elems[i]) != null; i++) {

 if (!keepData && elem.nodeType === 1) {
 jQuery.cleanData(getAll(elem));
 }

 if (elem.parentNode) {
 if (keepData && jQuery.contains(elem.ownerDocument,
elem)) {
 setGlobalEval(getAll(elem, "script"));
 }
 }
 }
 }
}

```

```

 elem.parentNode.removeChild(elem);
 }
}

return this;
},

empty: function() {
 var elem,
 i = 0;

 for (; (elem = this[i]) != null; i++) {
 // Remove element nodes and prevent memory leaks
 if (elem.nodeType === 1) {
 jQuery.cleanData(getAll(elem, false));
 }

 // Remove any remaining nodes
 while (elem.firstChild) {
 elem.removeChild(elem.firstChild);
 }

 // If this is a select, ensure that it displays empty (#12336)
 // Support: IE<9
 if (elem.options && jQuery.nodeName(elem, "select")) {
 elem.options.length = 0;
 }
 }

 return this;
},

clone: function(dataAndEvents, deepDataAndEvents) {
 dataAndEvents = dataAndEvents == null ? false : dataAndEvents;
 deepDataAndEvents = deepDataAndEvents == null ? dataAndEvents :
deepDataAndEvents;

 return this.map(function () {
 return jQuery.clone(this, dataAndEvents, deepDataAndEvents);
 });
},

html: function(value) {
 return jQuery.access(this, function(value) {
 var elem = this[0] || {};

```

```

 i = 0,
 l = this.length;

 if (value === undefined) {
 return elem.nodeType === 1 ?
 elem.innerHTML.replace(rinlinejQuery, "") :
 undefined;
 }

 // See if we can take a shortcut and just use innerHTML
 if (typeof value === "string" && !rnolnnerhtml.test(value) &&
 (jQuery.support.htmlSerialize || !rnoshimcache.test(value)))
&&
 (jQuery.support.leadingWhitespace || !
rleadingWhitespace.test(value)) &&
 !wrapMap[(rtagName.exec(value) || ["", ""])
[1].toLowerCase()]) {

 value = value.replace(rxhtmlTag, "<$1></$2>");

 try {
 for (; i < l; i++) {
 // Remove element nodes and prevent memory
leaks
 elem = this[i] || {};
 if (elem.nodeType === 1) {
 jQuery.cleanData(getAll(elem, false));
 elem.innerHTML = value;
 }
 }
 }

 elem = 0;

 // If using innerHTML throws an exception, use the fallback
method
 } catch(e) {}
 }

 if (elem) {
 this.empty().append(value);
 }
}, null, value, arguments.length);
},
replaceWith: function() {

```

```

var
 // Snapshot the DOM in case .domManip sweeps something relevant
into its fragment
 args = jQuery.map(this, function(elem) {
 return [elem.nextSibling, elem.parentNode];
 }),
 i = 0;

// Make the changes, replacing each context element with the new content
this.domManip(arguments, function(elem) {
 var next = args[i++],
 parent = args[i++];

 if (parent) {
 // Don't use the snapshot next if it has moved (#13810)
 if (next && next.parentNode !== parent) {
 next = this.nextSibling;
 }
 jQuery(this).remove();
 parent.insertBefore(elem, next);
 }
 // Allow new content to include elements from the context set
}, true);

// Force removal if there was no new content (e.g., from empty arguments)
return i ? this : this.remove();
},

detach: function(selector) {
 return this.remove(selector, true);
},

domManip: function(args, callback, allowIntersection) {

 // Flatten any nested arrays
 args = core_concat.apply([], args);

 var first, node, hasScripts,
 scripts, doc, fragment,
 i = 0,
 l = this.length,
 set = this,
 iNoClone = l - 1,
 value = args[0],
 isFunction = jQueryisFunction(value);
}

```

```

// We can't cloneNode fragments that contain checked, in WebKit
if (isFunction || !(i <= 1 || typeof value === "string" ||
jQuery.support.checkClone || !rchecked.test(value))) {
 return this.each(function(index) {
 var self = set.eq(index);
 if (isFunction) {
 args[0] = value.call(this, index, self.html());
 }
 self.domManip(args, callback, allowIntersection);
 });
}

if (!) {
 fragment = jQuery.buildFragment(args, this[0].ownerDocument,
false, !allowIntersection && this);
 first = fragment.firstChild;

 if (fragment.childNodes.length === 1) {
 fragment = first;
 }

 if (first) {
 scripts = jQuery.map(getAll(fragment, "script"), disableScript
);
 hasScripts = scripts.length;

 // Use the original fragment for the last item instead of the first
because it can end up
 // being emptied incorrectly in certain situations (#8070).
 for (; i < l; i++) {
 node = fragment;

 if (i !== iNoClone) {
 node = jQuery.clone(node, true, true);

 // Keep references to cloned scripts for later
restoration
 if (hasScripts) {
 jQuery.merge(scripts, getAll(node,
"script"));
 }
 }
 }
 }
}
callback.call(this[i], node, i);

```

```

 }

 if (hasScripts) {
 doc = scripts[scripts.length - 1].ownerDocument;

 // Reenable scripts
 jQuery.map(scripts, restoreScript);

 // Evaluate executable scripts on first document
insertion
 for (i = 0; i < hasScripts; i++) {
 node = scripts[i];
 if (rscriptType.test(node.type || "") &&
 !jQuery._data(node, "globalEval") &&
 jQuery.contains(doc, node)) {

 if (node.src) {
 // Hope ajax is available...
 jQuery._evalUrl(node.src);
 } else {
 jQuery.globalEval((node.text ||
node.textContent || node.innerHTML || "").replace(rcleanScript, ""));
 }
 }
 }

 // Fix #11809: Avoid leaking memory
 fragment = first = null;
 }
 }

 return this;
}
});

// Support: IE<8
// Manipulating tables requires a tbody
function manipulationTarget(elem, content) {
 return jQuery.nodeName(elem, "table") &&
 jQuery.nodeName(content.nodeType === 1 ? content : content.firstChild,
"tr") ?

 elem.getElementsByTagName("tbody")[0] ||
 elem.appendChild(elem.ownerDocument.createElement("tbody")) :

```

```

 elem;
 }

// Replace/restore the type attribute of script elements for safe DOM manipulation
function disableScript(elem) {
 elem.type = (jQuery.find.attr(elem, "type") !== null) + "/" + elem.type;
 return elem;
}
function restoreScript(elem) {
 var match = rscriptTypeMasked.exec(elem.type);
 if (match) {
 elem.type = match[1];
 } else {
 elem.removeAttribute("type");
 }
 return elem;
}

// Mark scripts as having already been evaluated
function setGlobalEval(elems, refElements) {
 var elem,
 i = 0;
 for (; (elem = elems[i]) != null; i++) {
 jQuery._data(elem, "globalEval", !refElements ||
 jQuery._data(refElements[i], "globalEval"));
 }
}

function cloneCopyEvent(src, dest) {

 if (dest.nodeType !== 1 || !jQuery.hasData(src)) {
 return;
 }

 var type, i, l,
 oldData = jQuery._data(src),
 curData = jQuery._data(dest, oldData),
 events = oldData.events;

 if (events) {
 delete curData.handle;
 curData.events = {};

 for (type in events) {
 for (i = 0, l = events[type].length; i < l; i++) {

```

```

 jQuery.event.add(dest, type, events[type][i]);
 }
}
}

// make the cloned public data object a copy from the original
if (curData.data) {
 curData.data = jQuery.extend({}, curData.data);
}
}

function fixCloneNodeIssues(src, dest) {
 var nodeName, e, data;

 // We do not need to do anything for non-Elements
 if (dest.nodeType !== 1) {
 return;
 }

 nodeName = dest.nodeName.toLowerCase();

 // IE6-8 copies events bound via attachEvent when using cloneNode.
 if (!jQuery.support.noCloneEvent && dest[jQuery.expando]) {
 data = jQuery._data(dest);

 for (e in data.events) {
 jQuery.removeEvent(dest, e, data.handle);
 }

 // Event data gets referenced instead of copied if the expando gets copied
 too
 dest.removeAttribute(jQuery.expando);
 }

 // IE blanks contents when cloning scripts, and tries to evaluate newly-set text
 if (nodeName === "script" && dest.text !== src.text) {
 disableScript(dest).text = src.text;
 restoreScript(dest);
 }

 // IE6-10 improperly clones children of object elements using classid.
 // IE10 throws NoModificationAllowedError if parent is null, #12132.
} else if (nodeName === "object") {
 if (dest.parentNode) {
 dest.outerHTML = src.outerHTML;
 }
}

```

```

// This path appears unavoidable for IE9. When cloning an object
// element in IE9, the outerHTML strategy above is not sufficient.
// If the src has innerHTML and the destination does not,
// copy the src.innerHTML into the dest.innerHTML. #10324
if (jQuery.support.html5Clone && (src.innerHTML && !
jQuery.trim(dest.innerHTML))) {
 dest.innerHTML = src.innerHTML;
}

} else if (nodeName === "input" && manipulation_rcheckableType.test(src.type))
{
 // IE6-8 fails to persist the checked state of a cloned checkbox
 // or radio button. Worse, IE6-7 fail to give the cloned element
 // a checked appearance if the defaultChecked value isn't also set

 dest.defaultChecked = dest.checked = src.checked;

 // IE6-7 get confused and end up setting the value of a cloned
 // checkbox/radio button to an empty string instead of "on"
 if (dest.value !== src.value) {
 dest.value = src.value;
 }

 // IE6-8 fails to return the selected option to the default selected
 // state when cloning options
} else if (nodeName === "option") {
 dest.defaultSelected = dest.selected = src.defaultSelected;

 // IE6-8 fails to set the defaultValue to the correct value when
 // cloning other types of input fields
} else if (nodeName === "input" || nodeName === "textarea") {
 dest.defaultValue = src.defaultValue;
}
}

jQuery.each({
 appendTo: "append",
 prependTo: "prepend",
 insertBefore: "before",
 insertAfter: "after",
 replaceAll: "replaceWith"
}, function(name, original) {
 jQuery.fn[name] = function(selector) {
 var elems,

```

```

 i = 0,
 ret = [],
 insert = jQuery(selector),
 last = insert.length - 1;

 for (; i <= last; i++) {
 elems = i === last ? this : this.clone(true);
 jQuery(insert[i])[original](elems);

 // Modern browsers can apply jQuery collections as arrays, but oldIE
needs a .get()
 core_push.apply(ret, elems.get());
 }

 return this.pushStack(ret);
 };
});

function getAll(context, tag) {
 var elems, elem,
 i = 0,
 found = typeof context.getElementsByTagName !== core_undefined ?
context.getElementsByTagName(tag || "*") :
 typeof context.querySelectorAll !== core_undefined ?
context.querySelectorAll(tag || "*") :
 undefined;

 if (!found) {
 for (found = [], elems = context.childNodes || context; (elem = elems[i]) != null; i++) {
 if (!tag || jQuery.nodeName(elem, tag)) {
 found.push(elem);
 } else {
 jQuery.merge(found, getAll(elem, tag));
 }
 }
 }

 return tag === undefined || tag && jQuery.nodeName(context, tag) ?
 jQuery.merge([context], found) :
 found;
}

// Used in buildFragment, fixes the defaultChecked property
function fixDefaultChecked(elem) {

```

```

if (manipulation_rcheckableType.test(elem.type)) {
 elem.defaultChecked = elem.checked;
}
}

jQuery.extend({
 clone: function(elem, dataAndEvents, deepDataAndEvents) {
 var destElements, node, clone, i, srcElements,
 inPage = jQuery.contains(elem.ownerDocument, elem);

 if (jQuery.support.html5Clone || jQuery.isXMLDoc(elem) || !
rnoshimcache.test("<" + elem.nodeName + ">")) {
 clone = elem.cloneNode(true);

 // IE<=8 does not properly clone detached, unknown element nodes
 } else {
 fragmentDiv.innerHTML = elem.outerHTML;
 fragmentDiv.removeChild(clone = fragmentDiv.firstChild);
 }

 if ((!jQuery.support.noCloneEvent || !jQuery.support.noCloneChecked) &&
 (elem.nodeType === 1 || elem.nodeType === 11) && !
jQuery.isXMLDoc(elem)) {

 // We eschew Sizzle here for performance reasons: http://jsperf.com/
getall-vs-sizzle/2
 destElements = getAll(clone);
 srcElements = getAll(elem);

 // Fix all IE cloning issues
 for (i = 0; (node = srcElements[i]) != null; ++i) {
 // Ensure that the destination node is not null; Fixes #9587
 if (destElements[i]) {
 fixCloneNodeIssues(node, destElements[i]);
 }
 }
 }

 // Copy the events from the original to the clone
 if (dataAndEvents) {
 if (deepDataAndEvents) {
 srcElements = srcElements || getAll(elem);
 destElements = destElements || getAll(clone);

 for (i = 0; (node = srcElements[i]) != null; i++) {

```

```

 cloneCopyEvent(node, destElements[i]);
 }
} else {
 cloneCopyEvent(elem, clone);
}
}

// Preserve script evaluation history
destElements = getAll(clone, "script");
if (destElements.length > 0) {
 setGlobalEval(destElements, !inPage && getAll(elem, "script"));
}

destElements = srcElements = node = null;

// Return the cloned set
return clone;
},

buildFragment: function(elems, context, scripts, selection) {
var j, elem, contains,
 tmp, tag, tbody, wrap,
 l = elems.length,

 // Ensure a safe fragment
 safe = createSafeFragment(context),

 nodes = [],
 i = 0;

for (; i < l; i++) {
 elem = elems[i];

 if (elem || elem === 0) {

 // Add nodes directly
 if (jQuery.type(elem) === "object") {
 jQuery.merge(nodes, elem.nodeType ? [elem] : elem);

 // Convert non-html into a text node
 } else if (!rhtml.test(elem)) {
 nodes.push(context.createTextNode(elem));

 // Convert html into DOM nodes
 } else {

```

```

tmp = tmp ||
safe.appendChild(context.createElement("div"));

// Deserialize a standard representation
tag = (rtagName.exec(elem) || ["", ""])
[1].toLowerCase();
wrap = wrapMap[tag] || wrapMap._default;

tmp.innerHTML = wrap[1] + elem.replace(rxhtmlTag,
"<$1></$2>") + wrap[2];

// Descend through wrappers to the right content
j = wrap[0];
while (j--) {
 tmp = tmp.lastChild;
}

// Manually add leading whitespace removed by IE
if (!jQuery.support.leadingWhitespace &&
rladingWhitespace.test(elem)) {

nodes.push(context.createTextNode(rleadingWhitespace.exec(elem)[0]));
}

// Remove IE's autoinserted <tbody> from table
fragments
if (!jQuery.support.tbody) {

 // String was a <table>, *may* have spurious
<tbody>
elem = tag === "table" && !rtbody.test(elem) ?
 tmp.firstChild :

 // String was a bare <thead> or <tfoot>
 wrap[1] === "<table>" && !
rtbody.test(elem) ?

 tmp :
 0;

 j = elem && elem.childNodes.length;
 while (j--) {
 if (jQuery.nodeName(tbody =
elem.childNodes[j]), "tbody") && !tbody.childNodes.length) {
 elem.removeChild(tbody);
 }
 }
}

```

```

 }
 }

 jQuery.merge(nodes, tmp.childNodes);

 // Fix #12392 for WebKit and IE > 9
 tmp.textContent = "";

 // Fix #12392 for oldIE
 while (tmp.firstChild) {
 tmp.removeChild(tmp.firstChild);
 }

 // Remember the top-level container for proper cleanup
 tmp = safe.lastChild;
}

}

// Fix #11356: Clear elements from fragment
if (tmp) {
 safe.removeChild(tmp);
}

// Reset defaultChecked for any radios and checkboxes
// about to be appended to the DOM in IE 6/7 (#8060)
if (!jQuery.support.appendChecked) {
 jQuery.grep(getAll(nodes, "input"), fixDefaultChecked);
}

i = 0;
while ((elem = nodes[i++])){

 // #4087 - If origin and destination elements are the same, and this is
 // that element, do not do anything
 if (selection && jQuery.inArray(elem, selection) !== -1) {
 continue;
 }

 contains = jQuery.contains(elem.ownerDocument, elem);

 // Append to fragment
 tmp = getAll(safe.appendChild(elem), "script");

 // Preserve script evaluation history
}

```

```

 if (contains) {
 setGlobalEval(tmp);
 }

 // Capture executables
 if (scripts) {
 j = 0;
 while ((elem = tmp[j++])) {
 if (rscriptType.test(elem.type || "")){
 scripts.push(elem);
 }
 }
 }
 }

 tmp = null;

 return safe;
},

cleanData: function(elems, /* internal */ acceptData) {
 var elem, type, id, data,
 i = 0,
 internalKey = jQuery.expando,
 cache = jQuery.cache,
 deleteExpando = jQuery.support.deleteExpando,
 special = jQuery.event.special;

 for (; (elem = elems[i]) != null; i++) {

 if (acceptData || jQuery.acceptData(elem)){

 id = elem[internalKey];
 data = id && cache[id];

 if (data){
 if (data.events){
 for (type in data.events){
 if (special[type]){
 jQuery.event.remove(elem, type);
 }
 }
 }
 // This is a shortcut to avoid
 jQuery.event.remove's overhead
 } else {
 }
 }
}

```

```

 jQuery.removeEvent(elem, type,
data.handle);
 }
 }
 }

 // Remove cache only if it was not already removed by
jQuery.event.remove
if (cache[id]) {

 delete cache[id];

 // IE does not allow us to delete expando
properties from nodes,
// nor does it have a removeAttribute function on
Document nodes;
 // we must handle all of these cases
 if (deleteExpando) {
 delete elem[internalKey];

 } else if (typeof elem.removeAttribute !==
core_undefined) {
 elem.removeAttribute(internalKey);

 } else {
 elem[internalKey] = null;
 }

 core_deletedIds.push(id);
}

}
}
}
},
}

_evalUrl: function(url) {
return jQuery.ajax({
 url: url,
 type: "GET",
 dataType: "script",
 async: false,
 global: false,
 "throws": true
});
}
}

```

```

});

jQuery.fn.extend({
 wrapAll: function(html) {
 if (jQuery.isFunction(html)) {
 return this.each(function(i) {
 jQuery(this).wrapAll(html.call(this, i));
 });
 }
 }

 if (this[0]) {
 // The elements to wrap the target around
 var wrap = jQuery(html, this[0].ownerDocument).eq(0).clone(true);

 if (this[0].parentNode) {
 wrap.insertBefore(this[0]);
 }

 wrap.map(function() {
 var elem = this;

 while (elem.firstChild && elem.firstChild.nodeType === 1) {
 elem = elem.firstChild;
 }

 return elem;
 }).append(this);
 }
}

return this;
},

wrapInner: function(html) {
 if (jQuery.isFunction(html)) {
 return this.each(function(i) {
 jQuery(this).wrapInner(html.call(this, i));
 });
 }
}

return this.each(function() {
 var self = jQuery(this),
 contents = self.contents();

 if (contents.length) {
 contents.wrapAll(html);
 }
}

```

```

 } else {
 self.append(html);
 }
 });
},
wrap: function(html) {
 varisFunction = jQuery.isFunction(html);

 return this.each(function(i) {
 jQuery(this).wrapAll(isFunction ? html.call(this, i) : html);
 });
},
unwrap: function() {
 return this.parent().each(function() {
 if (!jQuery.nodeName(this, "body")) {
 jQuery(this).replaceWith(this.childNodes);
 }
 }).end();
}
});
var iframe, getStyles, curCSS,
 ralpha = /alpha\([^\)]*\)/i,
 ropacity = /opacity\s*=\s*([^\)]*)/,
 rposition = /^((top|right|bottom|left)\$|),
 // swappable if display is none or starts with table except "table", "table-cell", or
 "table-caption"
 // see here for display values: https://developer.mozilla.org/en-US/docs/CSS/
 display
 rdisplayswap = /^((none|table(?![ea]).+|),
 rmargin = /^margin|,
 rnumsplit = new RegExp("^(" + core_pnum + ")(.*$)", "i"),
 rnumnonpx = new RegExp("^(" + core_pnum + ")(?!px)[a-z%]+$", "i"),
 rrelNum = new RegExp("^(+-)=(" + core_pnum + ")", "i"),
 elemdisplay = { BODY: "block" },

cssShow = { position: "absolute", visibility: "hidden", display: "block" },
cssNormalTransform = {
 letterSpacing: 0,
 fontWeight: 400
},
cssExpand = ["Top", "Right", "Bottom", "Left"],
cssPrefixes = ["Webkit", "O", "Moz", "ms"];

```

```

// return a css property mapped to a potentially vendor prefixed property
function vendorPropName(style, name) {

 // shortcut for names that are not vendor prefixed
 if (name in style) {
 return name;
 }

 // check for vendor prefixed names
 var capName = name.charAt(0).toUpperCase() + name.slice(1),
 origName = name,
 i = cssPrefixes.length;

 while (i--) {
 name = cssPrefixes[i] + capName;
 if (name in style) {
 return name;
 }
 }

 return origName;
}

function isHidden(elem, el) {
 // isHidden might be called from jQuery#filter function;
 // in that case, element will be second argument
 elem = el || elem;
 return jQuery.css(elem, "display") === "none" || !
jQuery.contains(elem.ownerDocument, elem);
}

function showHide(elements, show) {
 var display, elem, hidden,
 values = [],
 index = 0,
 length = elements.length;

 for (; index < length; index++) {
 elem = elements[index];
 if (!elem.style) {
 continue;
 }

 values[index] = jQuery._data(elem, "olddisplay");
}

```

```

display = elem.style.display;
if (show) {
 // Reset the inline display of this element to learn if it is
 // being hidden by cascaded rules or not
 if (!values[index] && display === "none") {
 elem.style.display = "";
 }

 // Set elements which have been overridden with display: none
 // in a stylesheet to whatever the default browser style is
 // for such an element
 if (elem.style.display === "" && isHidden(elem)) {
 values[index] = jQuery._data(elem, "olddisplay",
css_defaultDisplay(elem.nodeName));
 }
} else {

 if (!values[index]) {
 hidden = isHidden(elem);

 if (display && display !== "none" || !hidden) {
 jQuery._data(elem, "olddisplay", hidden ? display :
jQuery.css(elem, "display"));
 }
 }
}

// Set the display of most of the elements in a second loop
// to avoid the constant reflow
for (index = 0; index < length; index++) {
 elem = elements[index];
 if (!elem.style) {
 continue;
 }
 if (!show || elem.style.display === "none" || elem.style.display === "") {
 elem.style.display = show ? values[index] || "" : "none";
 }
}

return elements;
}

jQuery.fn.extend({
 css: function(name, value) {

```

```

 return jQuery.access(this, function(elem, name, value) {
 var len, styles,
 map = {},
 i = 0;

 if (jQuery.isArray(name)) {
 styles = getStyles(elem);
 len = name.length;

 for (; i < len; i++) {
 map[name[i]] = jQuery.css(elem, name[i], false,
styles);
 }
 }

 return map;
 }

 return value !== undefined ?
 jQuery.style(elem, name, value) :
 jQuery.css(elem, name);
 }, name, value, arguments.length > 1);
 },
 show: function() {
 return showHide(this, true);
 },
 hide: function() {
 return showHide(this);
 },
 toggle: function(state) {
 if (typeof state === "boolean") {
 return state ? this.show() : this.hide();
 }

 return this.each(function() {
 if (isHidden(this)) {
 jQuery(this).show();
 } else {
 jQuery(this).hide();
 }
 });
 }
});

jQuery.extend({
 // Add in style property hooks for overriding the default

```

```

// behavior of getting and setting a style property
cssHooks: {
 opacity: {
 get: function(elem, computed) {
 if (computed) {
 // We should always get a number back from opacity
 var ret = curCSS(elem, "opacity");
 return ret === "" ? "1" : ret;
 }
 }
 },
},
// Don't automatically add "px" to these possibly-unitless properties
cssNumber: {
 "columnCount": true,
 "fillOpacity": true,
 "fontWeight": true,
 "lineHeight": true,
 "opacity": true,
 "order": true,
 "orphans": true,
 "widows": true,
 "zIndex": true,
 "zoom": true
},
// Add in properties whose names you wish to fix before
// setting or getting the value
cssProps: {
 // normalize float css property
 "float": jQuery.support.cssFloat ? "cssFloat" : "styleFloat"
},
// Get and set the style property on a DOM Node
style: function(elem, name, value, extra) {
 // Don't set styles on text and comment nodes
 if (!elem || elem.nodeType === 3 || elem.nodeType === 8 || !elem.style) {
 return;
 }
 // Make sure that we're working with the right name
 var ret, type, hooks,
 origName = jQuery.camelCase(name),
 style = elem.style;
}

```

```

name = jQuery.cssProps[origName] || (jQuery.cssProps[origName] =
vendorPropName(style, origName));

// gets hook for the prefixed version
// followed by the unprefixed version
hooks = jQuery.cssHooks[name] || jQuery.cssHooks[origName];

// Check if we're setting a value
if (value !== undefined) {
 type = typeof value;

 // convert relative number strings (+= or -=) to relative numbers.

#7345
 if (type === "string" && (ret = rrelNum.exec(value))) {
 value = (ret[1] + 1) * ret[2] + parseFloat(jQuery.css(elem,
name));
 // Fixes bug #9237
 type = "number";
 }

 // Make sure that NaN and null values aren't set. See: #7116
 if (value == null || type === "number" && isNaN(value)) {
 return;
 }

 // If a number was passed in, add 'px' to the (except for certain CSS
properties)
 if (type === "number" && !jQuery.cssNumber[origName]) {
 value += "px";
 }

 // Fixes #8908, it can be done more correctly by specifying setters in
cssHooks,
 // but it would mean to define eight (for every problematic property)
identical functions
 if (!jQuery.support.clearCloneStyle && value === "" &&
name.indexOf("background") === 0) {
 style[name] = "inherit";
 }

 // If a hook was provided, use that value, otherwise just set the
specified value
 if (!hooks || !("set" in hooks) || (value = hooks.set(elem, value,
extra)) !== undefined) {

```

```

 // Wrapped to prevent IE from throwing errors when 'invalid'
values are provided
 // Fixes bug #5509
 try {
 style[name] = value;
 } catch(e) {}
 }

} else {
 // If a hook was provided get the non-computed value from there
 if (hooks && "get" in hooks && (ret = hooks.get(elem, false, extra)) !
== undefined) {
 return ret;
 }

 // Otherwise just get the value from the style object
 return style[name];
}
},
};

css: function(elem, name, extra, styles) {
 var num, val, hooks,
 origName = jQuery.camelCase(name);

 // Make sure that we're working with the right name
 name = jQuery.cssProps[origName] || (jQuery.cssProps[origName] =
vendorPropName(elem.style, origName));

 // gets hook for the prefixed version
 // followed by the unprefixed version
 hooks = jQuery.cssHooks[name] || jQuery.cssHooks[origName];

 // If a hook was provided get the computed value from there
 if (hooks && "get" in hooks) {
 val = hooks.get(elem, true, extra);
 }

 // Otherwise, if a way to get the computed value exists, use that
 if (val === undefined) {
 val = curCSS(elem, name, styles);
 }

 //convert "normal" to computed value
 if (val === "normal" && name in cssNormalTransform) {

```

```

 val = cssNormalTransform[name];
}

// Return, converting to number if forced or a qualifier was provided and val
looks numeric
if (extra === "" || extra) {
 num = parseFloat(val);
 return extra === true || jQuery.isNumeric(num) ? num || 0 : val;
}
return val;
};

// NOTE: we've included the "window" in window.getComputedStyle
// because jsdom on node.js will break without it.
if (window.getComputedStyle) {
 getStyles = function(elem) {
 return window.getComputedStyle(elem, null);
 };

 curCSS = function(elem, name, _computed) {
 var width, minWidth, maxWidth,
 computed = _computed || getStyles(elem),

 // getPropertyValue is only needed for .css('filter') in IE9, see #12537
 ret = computed ? computed.getPropertyValue(name) ||

computed[name] : undefined,
 style = elem.style;

 if (computed) {

 if (ret === "" && !jQuery.contains(elem.ownerDocument, elem)) {
 ret = jQuery.style(elem, name);
 }

 // A tribute to the "awesome hack by Dean Edwards"
 // Chrome < 17 and Safari 5.0 uses "computed value" instead of
"used value" for margin-right
 // Safari 5.1.7 (at least) returns percentage for a larger set of values,
but width seems to be reliably pixels
 // this is against the CSSOM draft spec: http://dev.w3.org/csswg/
cssom/#resolved-values
 if (rnumnonpx.test(ret) && rmargin.test(name)) {

 // Remember the original values

```

```

 width = style.width;
 minWidth = style.minWidth;
 maxWidth = style.maxWidth;

 // Put in the new values to get a computed value out
 style.minWidth = style.maxWidth = style.width = ret;
 ret = computed.width;

 // Revert the changed values
 style.width = width;
 style.minWidth = minWidth;
 style.maxWidth = maxWidth;
 }
}

return ret;
};

} else if (document.documentElement.currentStyle) {
 getStyles = function(elem) {
 return elem.currentStyle;
 };

 curCSS = function(elem, name, _computed) {
 var left, rs, rsLeft,
 computed = _computed || getStyles(elem),
 ret = computed ? computed[name] : undefined,
 style = elem.style;

 // Avoid setting ret to empty string here
 // so we don't default to auto
 if (ret == null && style && style[name]) {
 ret = style[name];
 }

 // From the awesome hack by Dean Edwards
 // http://erik.eae.net/archives/2007/07/27/18.54.15/#comment-102291

 // If we're not dealing with a regular pixel number
 // but a number that has a weird ending, we need to convert it to pixels
 // but not position css attributes, as those are proportional to the parent
element instead
 // and we can't measure the parent instead because it might trigger a
"stacking dolls" problem
 if (rnumnonpx.test(ret) && !rposition.test(name)) {

```

```

// Remember the original values
left = style.left;
rs = elem.runtimeStyle;
rsLeft = rs && rs.left;

// Put in the new values to get a computed value out
if (rsLeft) {
 rs.left = elem.currentStyle.left;
}
style.left = name === "fontSize" ? "1em" : ret;
ret = style.pixelLeft + "px";

// Revert the changed values
style.left = left;
if (rsLeft) {
 rs.left = rsLeft;
}
}

return ret === "" ? "auto" : ret;
};

}

function setPositiveNumber(elem, value, subtract) {
 var matches = rnumsplit.exec(value);
 return matches ?
 // Guard against undefined "subtract", e.g., when used as in cssHooks
 Math.max(0, matches[1] - (subtract || 0)) + (matches[2] || "px") :
 value;
}

function augmentWidthOrHeight(elem, name, extra, isBorderBox, styles) {
 var i = extra === (isBorderBox ? "border" : "content") ?
 // If we already have the right measurement, avoid augmentation
 4 :
 // Otherwise initialize for horizontal or vertical properties
 name === "width" ? 1 : 0,
 val = 0;

 for (; i < 4; i += 2) {
 // both box models exclude margin, so add it if we want it
 if (extra === "margin") {
 val += jQuery.css(elem, extra + cssExpand[i], true, styles);
 }
 }
}

```

```

if (isBorderBox) {
 // border-box includes padding, so remove it if we want content
 if (extra === "content") {
 val -= jQuery.css(elem, "padding" + cssExpand[i], true, styles
);
 }

 // at this point, extra isn't border nor margin, so remove border
 if (extra !== "margin") {
 val -= jQuery.css(elem, "border" + cssExpand[i] + "Width",
true, styles);
 }
} else {
 // at this point, extra isn't content, so add padding
 val += jQuery.css(elem, "padding" + cssExpand[i], true, styles);

 // at this point, extra isn't content nor padding, so add border
 if (extra !== "padding") {
 val += jQuery.css(elem, "border" + cssExpand[i] + "Width",
true, styles);
 }
}

return val;
}

function getWidthOrHeight(elem, name, extra) {

 // Start with offset property, which is equivalent to the border-box value
 var valuesBorderBox = true,
 val = name === "width" ? elem.offsetWidth : elem.offsetHeight,
 styles = getStyles(elem),
 isBorderBox = jQuery.support.boxSizing && jQuery.css(elem, "boxSizing",
false, styles) === "border-box";

 // some non-html elements return undefined for offsetWidth, so check for null/
undefined
 // svg - https://bugzilla.mozilla.org/show_bug.cgi?id=649285
 // MathML - https://bugzilla.mozilla.org/show_bug.cgi?id=491668
 if (val <= 0 || val == null) {
 // Fall back to computed then uncomputed css if necessary
 val = curCSS(elem, name, styles);
 if (val < 0 || val == null) {

```

```

 val = elem.style[name];
}

// Computed unit is not pixels. Stop here and return.
if (rnumnonpx.test(val)) {
 return val;
}

// we need the check for style in case a browser which returns unreliable
values
// for getComputedStyle silently falls back to the reliable elem.style
valueIsBorderBox = isBorderBox && (jQuery.support.boxSizingReliable || val
==== elem.style[name]);

// Normalize "", auto, and prepare for extra
val = parseFloat(val) || 0;
}

// use the active box-sizing model to add/subtract irrelevant styles
return (val +
 augmentWidthOrHeight(
 elem,
 name,
 extra || (isBorderBox ? "border" : "content"),
 valueIsBorderBox,
 styles
)
) + "px";
}

// Try to determine the default display value of an element
function css_defaultDisplay(nodeName) {
 var doc = document,
 display = elemdisplay[nodeName];

 if (!display) {
 display = actualDisplay(nodeName, doc);

 // If the simple way fails, read from inside an iframe
 if (display === "none" || !display) {
 // Use the already-created iframe if possible
 iframe = (iframe ||
 jQuery("<iframe frameborder='0' width='0' height='0'>")
 .css("cssText", "display:block !important")
).appendTo(doc.documentElement);
 }
 }
}

```

```

 // Always write a new HTML skeleton so Webkit and Firefox don't
choke on reuse
 doc = (iframe[0].contentWindow ||
iframe[0].contentDocument).document;
 doc.write("<!doctype html><html><body>");
 doc.close();

 display = actualDisplay(nodeName, doc);
 iframe.detach();
 }

 // Store the correct default display
 elemdisplay[nodeName] = display;
}

return display;
}

// Called ONLY from within css_defaultDisplay
function actualDisplay(name, doc) {
 var elem = jQuery(doc.createElement(name)).appendTo(doc.body),
 display = jQuery.css(elem[0], "display");
 elem.remove();
 return display;
}

jQuery.each(["height", "width"], function(i, name) {
 jQuery.cssHooks[name] = {
 get: function(elem, computed, extra) {
 if (computed) {
 // certain elements can have dimension info if we invisibly
show them
 // however, it must have a current display style that would
benefit from this
 return elem.offsetWidth === 0 &&
rdisplayswap.test(jQuery.css(elem, "display")) ?
 jQuery.swap(elem, cssShow, function() {
 return getWidthOrHeight(elem, name, extra);
 } :
 getWidthOrHeight(elem, name, extra);
 }
 },
 set: function(elem, value, extra) {

```

```

var styles = extra && getStyles(elem);
return setPositiveNumber(elem, value, extra ?
 augmentWidthOrHeight(
 elem,
 name,
 extra,
 jQuery.support.boxSizing && jQuery.css(elem,
"boxSizing", false, styles) === "border-box",
 styles
) : 0
);
}
};

if (!jQuery.support.opacity) {
 jQuery.cssHooks.opacity = {
 get: function(elem, computed) {
 // IE uses filters for opacity
 return ropacity.test((computed && elem.currentStyle ?
elem.currentStyle.filter : elem.style.filter) || "") ?
 (0.01 * parseFloat(RegExp.$1)) + "" :
 computed ? "1" : "";
 },
 set: function(elem, value) {
 var style = elem.style,
 currentStyle = elem.currentStyle,
 opacity = jQuery.isNumeric(value) ? "alpha(opacity=" + value
* 100 + ")" : "",
 filter = currentStyle && currentStyle.filter || style.filter || "";
 // IE has trouble with opacity if it does not have layout
 // Force it by setting the zoom level
 style.zoom = 1;

 // if setting opacity to 1, and no other filters exist - attempt to remove
filter attribute #6652
 // if value === "", then remove inline opacity #12685
 if ((value >= 1 || value === "") &&
 jQuery.trim(filter.replace(ralpha, "")) === "" &&
 style.removeAttribute) {

 // Setting style.filter to null, "" & " " still leave "filter:" in the
cssText
 }
 }
 };
}

```

```

 // if "filter:" is present at all, clearType is disabled, we want to
avoid this
 // style.removeAttribute is IE Only, but so apparently is this
code path...
 style.removeAttribute("filter");

 // if there is no filter style applied in a css rule or unset inline
opacity, we are done
 if (value === "" || currentStyle && !currentStyle.filter) {
 return;
 }
 }

 // otherwise, set new filter values
 style.filter = ralpha.test(filter) ?
 filter.replace(ralpha, opacity) :
 filter + " " + opacity;
}
};

}

// These hooks cannot be added until DOM ready because the support test
// for it is not run until after DOM ready
jQuery(function() {
 if (!jQuery.support.reliableMarginRight) {
 jQuery.cssHooks.marginRight = {
 get: function(elem, computed) {
 if (computed) {
 // WebKit Bug 13343 - getComputedStyle returns
wrong value for margin-right
 // Work around by temporarily setting element display
to inline-block
 return jQuery.swap(elem, { "display": "inline-block" },
curCSS, [elem, "marginRight"]);
 }
 }
 };
 }

 // Webkit bug: https://bugs.webkit.org/show_bug.cgi?id=29084
 // getComputedStyle returns percent when specified for top/left/bottom/right
 // rather than make the css module depend on the offset module, we just check for
it here
 if (!jQuery.support.pixelPosition && jQuery.fn.position) {
 jQuery.each(["top", "left"], function(i, prop) {

```

```

jQuery.cssHooks[prop] = {
 get: function(elem, computed) {
 if (computed) {
 computed = curCSS(elem, prop);
 // if curCSS returns percentage, fallback to offset
 return rnumnonpx.test(computed) ?
 jQuery(elem).position()[prop] + "px" :
 computed;
 }
 }
};

});

if (jQuery.expr && jQuery.expr.filters) {
 jQuery.expr.filters.hidden = function(elem) {
 // Support: Opera <= 12.12
 // Opera reports offsetWidths and offsetHeights less than zero on some
elements
 return elem.offsetWidth <= 0 && elem.offsetHeight <= 0 ||
 (!jQuery.support.reliableHiddenOffsets && ((elem.style &&
elem.style.display) || jQuery.css(elem, "display") === "none"));
 };

 jQuery.expr.filters.visible = function(elem) {
 return !jQuery.expr.filters.hidden(elem);
 };
}

// These hooks are used by animate to expand properties
jQuery.each({
 margin: "",
 padding: "",
 border: "Width"
}, function(prefix, suffix) {
 jQuery.cssHooks[prefix + suffix] = {
 expand: function(value) {
 var i = 0,
 expanded = {},
 // assumes a single number if not a string
 parts = typeof value === "string" ? value.split(" ") : [value];

```



```

 return { name: elem.name, value:
val.replace(rCRLF, "\r\n") };
 } :
 { name: elem.name, value: val.replace(rCRLF, "\r\n") };
}).get();
}
});

//Serialize an array of form elements or a set of
//key/values into a query string
jQuery.param = function(a, traditional) {
 var prefix,
 s = [],
 add = function(key, value) {
 // If value is a function, invoke it and return its value
 value = jQueryisFunction(value) ? value() : (value == null ? "" : value
);
 s[s.length] = encodeURIComponent(key) + "=" +
encodeURIComponent(value);
 };

 // Set traditional to true for jQuery <= 1.3.2 behavior.
 if (traditional === undefined) {
 traditional = jQuery.ajaxSettings && jQuery.ajaxSettings.traditional;
 }

 // If an array was passed in, assume that it is an array of form elements.
 if (jQuery.isArray(a) || (a.jquery && !jQuery.isPlainObject(a))) {
 // Serialize the form elements
 jQuery.each(a, function() {
 add(this.name, this.value);
 });
 }

} else {
 // If traditional, encode the "old" way (the way 1.3.2 or older
 // did it), otherwise encode params recursively.
 for (prefix in a) {
 buildParams(prefix, a[prefix], traditional, add);
 }
}

// Return the resulting serialization
return s.join("&").replace(r20, "+");
};

```

```

function buildParams(prefix, obj, traditional, add) {
 var name;

 if (jQuery.isArray(obj)) {
 // Serialize array item.
 jQuery.each(obj, function(i, v) {
 if (traditional || rbracket.test(prefix)) {
 // Treat each array item as a scalar.
 add(prefix, v);
 }
 });
 } else {
 // Item is non-scalar (array or object), encode its numeric index.
 buildParams(prefix + "[" + (typeof v === "object" ? i : "") +
 "] ", v, traditional, add);
 }
}

} else if (!traditional && jQuery.type(obj) === "object") {
 // Serialize object item.
 for (name in obj) {
 buildParams(prefix + "[" + name + "]", obj[name], traditional, add);
 }
}

} else {
 // Serialize scalar item.
 add(prefix, obj);
}
}

jQuery.each(("blur focus focusin focusout load resize scroll unload click dblclick " +
 "mousedown mouseup mousemove mouseover mouseout mouseenter
mouseleave " +
 "change select submit keydown keypress keyup error contextmenu").split(" "), function(i, name) {

 // Handle event binding
 jQuery.fn[name] = function(data, fn) {
 return arguments.length > 0 ?
 this.on(name, null, data, fn) :
 this.trigger(name);
 };
});

jQuery.fn.extend({
 hover: function(fnOver, fnOut) {
 return this.mouseenter(fnOver).mouseleave(fnOut || fnOver);
 }
});

```

```

 },
 bind: function(types, data, fn) {
 return this.on(types, null, data, fn);
 },
 unbind: function(types, fn) {
 return this.off(types, null, fn);
 },
 delegate: function(selector, types, data, fn) {
 return this.on(types, selector, data, fn);
 },
 undelegate: function(selector, types, fn) {
 // (namespace) or (selector, types [, fn])
 return arguments.length === 1 ? this.off(selector, "***") : this.off(types,
selector || "***", fn);
 }
});
var
 // Document location
 ajaxLocParts,
 ajaxLocation,
 ajax_nonce = jQuery.now(),

 ajax_rquery = /\?/,
 rhash = /#.*/$|,
 rts = /(?:&|=|^&)*|,
 rheaders = /(?:^|[\t\r\n]+)([\r\n]+|$)/mg, // IE leaves an \r character at EOL
 // #7653, #8125, #8152: local protocol detection
 rlocalProtocol = /(?:about|app-storage|file|reslwidget):$/,
 rnoContent = /(?:GET|HEAD)$|,
 rprotocol = /[^/]+/,
 rurl = /(?:^|[\w\.-]+:)(?:\/\/|([^\?#]*)(?:|(\d+))|)/,

 // Keep a copy of the old load method
 _load = jQuery.fn.load,

 /* Prefilters
 * 1) They are useful to introduce custom dataTypes (see ajax/jsonp.js for an
example)
 * 2) These are called:
 * - BEFORE asking for a transport
 * - AFTER param serialization (s.data is a string if s.processData is true)
 * 3) key is the dataType
 * 4) the catchall symbol "*" can be used

```

```

* 5) execution will start with transport dataType and THEN continue down to "*" if
needed
*/
prefilters = {},

/* Transports bindings
 * 1) key is the dataType
 * 2) the catchall symbol "*" can be used
 * 3) selection will start with transport dataType and THEN go to "*" if needed
*/
transports = {},

// Avoid comment-prolog char sequence (#10098); must appease lint and evade
compression
allTypes = "*/".concat("*");

// #8138, IE may throw an exception when accessing
// a field from window.location if document.domain has been set
try {
 ajaxLocation = location.href;
} catch(e) {
 // Use the href attribute of an A element
 // since IE will modify it given document.location
 ajaxLocation = document.createElement("a");
 ajaxLocation.href = "";
 ajaxLocation = ajaxLocation.href;
}

// Segment location into parts
ajaxLocParts = rurl.exec(ajaxLocation.toLowerCase()) || [];

// Base "constructor" for jQuery.ajaxPrefilter and jQuery.ajaxTransport
function addToPrefiltersOrTransports(structure) {

 // dataTypeExpression is optional and defaults to "*"
 return function(dataTypeExpression, func) {

 if (typeof dataTypeExpression !== "string") {
 func = dataTypeExpression;
 dataTypeExpression = "*";
 }

 var dataType,
 i = 0,

```

```

 dataTypes =
dataTypeExpression.toLowerCase().match(core_rnotwhite) || [];

 if (jQueryisFunction(func)) {
 // For each dataType in the dataTypeExpression
 while ((dataType = dataTypes[i++])) {
 // Prepend if requested
 if (dataType[0] === "+") {
 dataType = dataType.slice(1) || "*";
 (structure[dataType] = structure[dataType] || []
[]).unshift(func);
 }
 // Otherwise append
 } else {
 (structure[dataType] = structure[dataType] || []
[]).push(func);
 }
 }
 };

// Base inspection function for prefilters and transports
function inspectPrefiltersOrTransports(structure, options, originalOptions, jqXHR) {

 var inspected = {},
 seekingTransport = (structure === transports);

 function inspect(dataType) {
 var selected;
 inspected[dataType] = true;
 jQuery.each(structure[dataType] || [], function(_, prefilterOrFactory) {
 var dataTypeOrTransport = prefilterOrFactory(options,
originalOptions, jqXHR);
 if(typeof dataTypeOrTransport === "string" && !seekingTransport
&& !inspected[dataTypeOrTransport]) {
 options.dataTypes.unshift(dataTypeOrTransport);
 inspect(dataTypeOrTransport);
 return false;
 } else if (seekingTransport) {
 return !(selected = dataTypeOrTransport);
 }
 });
 return selected;
 }
}

```

```

return inspect(options.dataTypes[0]) || !inspected["*"] && inspect("*");
}

// A special extend for ajax options
// that takes "flat" options (not to be deep extended)
// Fixes #9887
function ajaxExtend(target, src) {
 var deep, key,
 flatOptions = jQuery.ajaxSettings.flatOptions || {};
 for (key in src) {
 if (src[key] !== undefined) {
 (flatOptions[key] ? target : (deep || (deep = {}))[key]) = src[key];
 }
 }
 if (deep) {
 jQuery.extend(true, target, deep);
 }
 return target;
}

jQuery.fn.load = function(url, params, callback) {
 if (typeof url !== "string" && _load) {
 return _load.apply(this, arguments);
 }

 var selector, response, type,
 self = this,
 off = url.indexOf(" ");
 if (off >= 0) {
 selector = url.slice(off, url.length);
 url = url.slice(0, off);
 }

 // If it's a function
 if (jQueryisFunction(params)) {

 // We assume that it's the callback
 callback = params;
 params = undefined;

 // Otherwise, build a param string
 }
}

```

```

} else if (params && typeof params === "object") {
 type = "POST";
}

// If we have elements to modify, make the request
if (self.length > 0) {
 jQuery.ajax({
 url: url,

 // if "type" variable is undefined, then "GET" method will be used
 type: type,
 dataType: "html",
 data: params
 }).done(function(responseText) {

 // Save response for use in complete callback
 response = arguments;

 self.html(selector ?

 // If a selector was specified, locate the right elements in a
 dummy div
 // Exclude scripts to avoid IE 'Permission Denied' errors

 jQuery("<div>").append(jQuery.parseHTML(responseText)).find(selector) :

 // Otherwise use the full result
 responseText);

 }).complete(callback && function(jqXHR, status) {
 self.each(callback, response || [jqXHR.responseText, status, jqXHR]);
 });
}

return this;
};

// Attach a bunch of functions for handling common AJAX events
jQuery.each(["ajaxStart", "ajaxStop", "ajaxComplete", "ajaxError", "ajaxSuccess",
"ajaxSend"], function(i, type){
 jQuery.fn[type] = function(fn){
 return this.on(type, fn);
 };
});

```

```
jQuery.extend({

 // Counter for holding the number of active queries
 active: 0,

 // Last-Modified header cache for next request
 lastModified: {},
 etag: {},

 ajaxSettings: {
 url: ajaxLocation,
 type: "GET",
 isLocal: rlocalProtocol.test(ajaxLocParts[1]),
 global: true,
 processData: true,
 async: true,
 contentType: "application/x-www-form-urlencoded; charset=UTF-8",
 /*
 timeout: 0,
 data: null,
 dataType: null,
 username: null,
 password: null,
 cache: null,
 throws: false,
 traditional: false,
 headers: {},
 */

 accepts: {
 "*": allTypes,
 text: "text/plain",
 html: "text/html",
 xml: "application/xml, text/xml",
 json: "application/json, text/javascript"
 },

 contents: {
 xml: /xml/,
 html: /html/,
 json: /json/
 },

 responseFields: {
 xml: "responseXML",
 }
 }
});
```

```

 text: "responseText",
 json: "responseJSON"
 },
}

// Data converters
// Keys separate source (or catchall "*") and destination types with a single
space
converters: {

 // Convert anything to text
 "* text": String,

 // Text to html (true = no transformation)
 "text html": true,

 // Evaluate text as a json expression
 "text json": jQuery.parseJSON,

 // Parse text as xml
 "text xml": jQuery.parseXML
},
}

// For options that shouldn't be deep extended:
// you can add your own custom options here if
// and when you create one that shouldn't be
// deep extended (see ajaxExtend)
flatOptions: {
 url: true,
 context: true
}
},

// Creates a full fledged settings object into target
// with both ajaxSettings and settings fields.
// If target is omitted, writes into ajaxSettings.
ajaxSetup: function(target, settings) {
 return settings ?

 // Building a settings object
 ajaxExtend(ajaxExtend(target, jQuery.ajaxSettings), settings) :

 // Extending ajaxSettings
 ajaxExtend(jQuery.ajaxSettings, target);
},

```

```

ajaxPrefilter: addToPrefiltersOrTransports(prefilters),
ajaxTransport: addToPrefiltersOrTransports(transports),

// Main method
ajax: function(url, options) {

 // If url is an object, simulate pre-1.5 signature
 if (typeof url === "object") {
 options = url;
 url = undefined;
 }

 // Force options to be an object
 options = options || {};

 var // Cross-domain detection vars
 parts,
 // Loop variable
 i,
 // URL without anti-cache param
 cacheURL,
 // Response headers as string
 responseHeadersString,
 // timeout handle
 timeoutTimer,

 // To know if global events are to be dispatched
 fireGlobals,

 transport,
 // Response headers
 responseHeaders,
 // Create the final options object
 s = jQuery.ajaxSetup({}, options),
 // Callbacks context
 callbackContext = s.context || s,
 // Context for global events is callbackContext if it is a DOM node or
jQuery collection
 globalEventContext = s.context && (callbackContext.nodeType || callbackContext.jquery) ?
 jQuery(callbackContext) :
 jQuery.event,
 // Deferreds
 deferred = jQuery.Deferred(),
 completeDeferred = jQuery.Callbacks("once memory"),

```

```

// Status-dependent callbacks
statusCode = s.statusCode || {},
// Headers (they are sent all at once)
requestHeaders = {},
requestHeadersNames = {},
// The jqXHR state
state = 0,
// Default abort message
strAbort = "canceled",
// Fake xhr
jqXHR = {
 readyState: 0,

 // Builds headers hashtable if needed
 getResponseHeader: function(key) {
 var match;
 if (state === 2) {
 if (!responseHeaders) {
 responseHeaders = {};
 while ((match =
rheaders.exec(responseHeadersString))) {

responseHeaders[match[1].toLowerCase()] = match[2];
}
}
match = responseHeaders[key.toLowerCase()];
}
return match == null ? null : match;
},

// Raw string
getAllResponseHeaders: function() {
 return state === 2 ? responseHeadersString : null;
},

// Caches the header
setRequestHeader: function(name, value) {
 var lname = name.toLowerCase();
 if (!state) {
 name = requestHeadersNames[lname] =
requestHeadersNames[lname] || name;
 requestHeaders[name] = value;
 }
 return this;
},

```

```

// Overrides response content-type header
overrideMimeType: function(type) {
 if (!state) {
 s.mimeType = type;
 }
 return this;
},

// Status-dependent callbacks
statusCode: function(map) {
 var code;
 if (map) {
 if (state < 2) {
 for (code in map) {
 // Lazy-add the new callback in a
way that preserves old ones
 statusCode[code] =
[statusCode[code], map[code]];
 }
 } else {
 // Execute the appropriate callbacks
 jqXHR.always(map[jqXHR.status]);
 }
 }
 return this;
},

// Cancel the request
abort: function(statusText) {
 var finalText = statusText || strAbort;
 if (transport) {
 transport.abort(finalText);
 }
 done(0, finalText);
 return this;
}
};

// Attach deferreds
deferred.promise(jqXHR).complete = completeDeferred.add;
jqXHR.success = jqXHR.done;
jqXHR.error = jqXHR.fail;

// Remove hash character (#7531: and string promotion)

```

```

// Add protocol if not provided (#5866: IE7 issue with protocol-less urls)
// Handle falsy url in the settings object (#10093: consistency with old
signature)
 // We also use the url parameter if available
 s.url = ((url || s.url || ajaxLocation) + "").replace(rhash,
"").replace(rprotocol, ajaxLocParts[1] + "//");

 // Alias method option to type as per ticket #12004
 s.type = options.method || options.type || s.method || s.type;

 // Extract dataTypes list
 s.dataTypes = jQuery.trim(s.dataType ||
"**").toLowerCase().match(core_rnotwhite) || [""];

 // A cross-domain request is in order when we have a protocol:host:port
mismatch
 if (s.crossDomain == null) {
 parts = rurl.exec(s.url.toLowerCase());
 s.crossDomain = !(parts &&
 (parts[1] !== ajaxLocParts[1] || parts[2] !== ajaxLocParts[2]
||

 (parts[3] || (parts[1] === "http:" ? "80" : "443")) !
==

 (ajaxLocParts[3] || (ajaxLocParts[1] === "http:"
? "80" : "443"))
);
 }

 // Convert data if not already a string
 if (s.data && s.processData && typeof s.data !== "string") {
 s.data = jQuery.param(s.data, s.traditional);
 }

 // Apply prefilters
 inspectPrefiltersOrTransports(prefilters, s, options, jqXHR);

 // If request was aborted inside a prefilter, stop there
 if (state === 2) {
 return jqXHR;
 }

 // We can fire global events as of now if asked to
 fireGlobals = s.global;

 // Watch for a new set of requests

```

```

if (fireGlobals && jQuery.active++ === 0) {
 jQuery.event.trigger("ajaxStart");
}

// Uppercase the type
s.type = s.type.toUpperCase();

// Determine if request has content
s.hasContent = !rnoContent.test(s.type);

// Save the URL in case we're toying with the If-Modified-Since
// and/or If-None-Match header later on
cacheURL = s.url;

// More options handling for requests with no content
if (!s.hasContent) {

 // If data is available, append data to url
 if (s.data) {
 cacheURL = (s.url += (ajax_rquery.test(cacheURL) ? "&" : "?")
) + s.data);
 // #9682: remove data so that it's not used in an eventual retry
 delete s.data;
 }

 // Add anti-cache in url if needed
 if (s.cache === false) {
 s.url = rts.test(cacheURL) ?

 // If there is already a '_' parameter, set its value
 cacheURL.replace(rts, "$1_=" + ajax_nonce++) :

 // Otherwise add one to the end
 cacheURL + (ajax_rquery.test(cacheURL) ? "&" : "?") +
"_" + ajax_nonce++;
 }
}

// Set the If-Modified-Since and/or If-None-Match header, if in ifModified
mode.
if (s.ifModified) {
 if (jQuery.lastModified[cacheURL]) {
 jqXHR.setRequestHeader("If-Modified-Since",
jQuery.lastModified[cacheURL]);
 }
}

```

```

 if (jQuery.etag[cacheURL]) {
 jqXHR.setRequestHeader("If-None-Match",
jQuery.etag[cacheURL]);
 }
 }

 // Set the correct header, if data is being sent
 if (s.data && s.hasContent && s.contentType !== false ||
options.contentType) {
 jqXHR.setRequestHeader("Content-Type", s.contentType);
 }

 // Set the Accepts header for the server, depending on the dataType
 jqXHR.setRequestHeader(
 "Accept",
 s.dataTypes[0] && s.accepts[s.dataTypes[0]] ?
 s.accepts[s.dataTypes[0]] + (s.dataTypes[0] !== "*" ? ", " +
allTypes + "; q=0.01" : "") :
 s.accepts["*"]
);

 // Check for headers option
 for (i in s.headers) {
 jqXHR.setRequestHeader(i, s.headers[i]);
 }

 // Allow custom headers/mimetypes and early abort
 if (s.beforeSend && (s.beforeSend.call(callbackContext, jqXHR, s) ===
false || state === 2)) {
 // Abort if not done already and return
 return jqXHR.abort();
 }

 // aborting is no longer a cancellation
 strAbort = "abort";

 // Install callbacks on deferreds
 for (i in { success: 1, error: 1, complete: 1 }) {
 jqXHR[i](s[i]);
 }

 // Get transport
 transport = inspectPrefiltersOrTransports(transports, s, options, jqXHR);

 // If no transport, we auto-abort

```

```

if (!transport) {
 done(-1, "No Transport");
} else {
 jqXHR.readyState = 1;

 // Send global event
 if (fireGlobals) {
 globalEventContext.trigger("ajaxSend", [jqXHR, s]);
 }
 // Timeout
 if (s.async && s.timeout > 0) {
 timeoutTimer = setTimeout(function() {
 jqXHR.abort("timeout");
 }, s.timeout);
 }

 try {
 state = 1;
 transport.send(requestHeaders, done);
 } catch (e) {
 // Propagate exception as error if not done
 if (state < 2) {
 done(-1, e);
 }
 // Simply rethrow otherwise
 } else {
 throw e;
 }
 }
}

// Callback for when everything is done
function done(status, nativeStatusText, responses, headers) {
 var isSuccess, success, error, response, modified,
 statusText = nativeStatusText;

 // Called once
 if (state === 2) {
 return;
 }

 // State is "done" now
 state = 2;

 // Clear timeout if it exists
 if (timeoutTimer) {

```

```

 clearTimeout(timeoutTimer);
 }

 // Dereference transport for early garbage collection
 // (no matter how long the jqXHR object will be used)
 transport = undefined;

 // Cache response headers
 responseHeadersString = headers || "";

 // Set readyState
 jqXHR.readyState = status > 0 ? 4 : 0;

 // Determine if successful
 isSuccess = status >= 200 && status < 300 || status === 304;

 // Get response data
 if (responses) {
 response = ajaxHandleResponses(s, jqXHR, responses);
 }

 // Convert no matter what (that way responseXXX fields are always
set)
 response = ajaxConvert(s, response, jqXHR, isSuccess);

 // If successful, handle type chaining
 if (isSuccess) {

 // Set the If-Modified-Since and/or If-None-Match header, if in
ifModified mode.
 if (s.ifModified) {
 modified = jqXHR.getResponseHeader("Last-
Modified");
 if (modified) {
 jQuery.lastModified[cacheURL] = modified;
 }
 modified = jqXHR.getResponseHeader("etag");
 if (modified) {
 jQuery.etag[cacheURL] = modified;
 }
 }

 // if no content
 if (status === 204 || s.type === "HEAD") {
 statusText = "nocontent";
 }
 }
}

```

```

 // if not modified
 } else if (status === 304) {
 statusText = "notmodified";

 // If we have data, let's convert it
 } else {
 statusText = response.state;
 success = response.data;
 error = response.error;
 isSuccess = !error;
 }
} else {
 // We extract error from statusText
 // then normalize statusText and status for non-aborts
 error = statusText;
 if (status || !statusText) {
 statusText = "error";
 if (status < 0) {
 status = 0;
 }
 }
}

// Set data for the fake xhr object
jqXHR.status = status;
jqXHR.statusText = (nativeStatusText || statusText) + "";

// Success/Error
if (isSuccess) {
 deferred.resolveWith(callbackContext, [success, statusText,
jqXHR]);
} else {
 deferred.rejectWith(callbackContext, [jqXHR, statusText, error
]);
}

// Status-dependent callbacks
jqXHR.statusCode(statusCode);
statusCode = undefined;

if (fireGlobals) {
 globalEventContext.trigger(isSuccess ? "ajaxSuccess" :
"ajaxError",
"ajaxError",
[jqXHR, s, isSuccess ? success : error]);
}

```

```

 }

 // Complete
 completeDeferred.fireWith(callbackContext, [jqXHR, statusText]);

 if (fireGlobals) {
 globalEventContext.trigger("ajaxComplete", [jqXHR, s]);
 // Handle the global AJAX counter
 if (!(--jQuery.active)) {
 jQuery.event.trigger("ajaxStop");
 }
 }
 }

 return jqXHR;
},
getJSON: function(url, data, callback) {
 return jQuery.get(url, data, callback, "json");
},
getScript: function(url, callback) {
 return jQuery.get(url, undefined, callback, "script");
}
});

jQuery.each(["get", "post"], function(i, method) {
 jQuery[method] = function(url, data, callback, type) {
 // shift arguments if data argument was omitted
 if (jQuery.isFunction(data)) {
 type = type || callback;
 callback = data;
 data = undefined;
 }

 return jQuery.ajax({
 url: url,
 type: method,
 dataType: type,
 data: data,
 success: callback
 });
 };
});

```

```

/* Handles responses to an ajax request:
 * - finds the right dataType (mediates between content-type and expected dataType)
 * - returns the corresponding response
 */
function ajaxHandleResponses(s, jqXHR, responses) {
 var firstDataType, ct, finalDataType, type,
 contents = s.contents,
 dataTypes = s.dataTypes;

 // Remove auto dataType and get content-type in the process
 while(dataTypes[0] === "*") {
 dataTypes.shift();
 if (ct === undefined) {
 ct = s.mimeType || jqXHR.getResponseHeader("Content-Type");
 }
 }

 // Check if we're dealing with a known content-type
 if (ct) {
 for (type in contents) {
 if (contents[type] && contents[type].test(ct)) {
 dataTypes.unshift(type);
 break;
 }
 }
 }

 // Check to see if we have a response for the expected dataType
 if (dataTypes[0] in responses) {
 finalDataType = dataTypes[0];
 } else {
 // Try convertible dataTypes
 for (type in responses) {
 if (!dataTypes[0] || s.converters[type + " " + dataTypes[0]]) {
 finalDataType = type;
 break;
 }
 if (!firstDataType) {
 firstDataType = type;
 }
 }
 // Or just use first one
 finalDataType = finalDataType || firstDataType;
 }
}

```

```

// If we found a dataType
// We add the dataType to the list if needed
// and return the corresponding response
if (finalDataType) {
 if (finalDataType !== dataTypes[0]) {
 dataTypes.unshift(finalDataType);
 }
 return responses[finalDataType];
}
}

/* Chain conversions given the request and the original response
 * Also sets the responseXXX fields on the jqXHR instance
 */
function ajaxConvert(s, response, jqXHR, isSuccess) {
 var conv2, current, conv, tmp, prev,
 converters = {},
 // Work with a copy of dataTypes in case we need to modify it for conversion
 dataTypes = s.dataTypes.slice();

 // Create converters map with lowercased keys
 if (dataTypes[1]) {
 for (conv in s.converters) {
 converters[conv.toLowerCase()] = s.converters[conv];
 }
 }

 current = dataTypes.shift();

 // Convert to each sequential dataType
 while (current) {

 if (s.responseFields[current]) {
 jqXHR[s.responseFields[current]] = response;
 }

 // Apply the dataFilter if provided
 if (!prev && isSuccess && s.dataFilter) {
 response = s.dataFilter(response, s.dataType);
 }

 prev = current;
 current = dataTypes.shift();

 if (current) {

```

```

// There's only work to do if current dataType is non-auto
if (current === "*") {

 current = prev;

 // Convert response if prev dataType is non-auto and differs from
current
} else if (prev !== "*" && prev !== current) {

 // Seek a direct converter
 conv = converters[prev + " " + current] || converters["* " +
current];

 // If none found, seek a pair
 if (!conv) {
 for (conv2 in converters) {

 // If conv2 outputs current
 tmp = conv2.split(" ");
 if (tmp[1] === current) {

 // If prev can be converted to accepted
input
 conv = converters[prev + " " + tmp[0]] ||
 converters["* " + tmp[0]];
 if (conv) {
 // Condense equivalence converters
 if (conv === true) {
 conv = converters[conv2];

 // Otherwise, insert the
intermediate dataType
 } else if (converters[conv2] !==
true) {
 current = tmp[0];
 dataTypes.unshift(tmp[1]);
 }
 break;
 }
 }
 }
 }
}

// Apply converter (if not an equivalence)

```

```

 if (conv !== true) {

 // Unless errors are allowed to bubble, catch and return
 them

 if (conv && s["throws"]) {
 response = conv(response);
 } else {
 try {
 response = conv(response);
 } catch (e) {
 return { state: "parsererror", error: conv ? e
: "No conversion from " + prev + " to " + current };
 }
 }
 }

 return { state: "success", data: response };
 }
// Install script dataType
jQuery.ajaxSetup({
 accepts: {
 script: "text/javascript, application/javascript, application/ecmascript,
application/x-ecmascript"
 },
 contents: {
 script: /(?:javalecma)script/
 },
 converters: {
 "text script": function(text) {
 jQuery.globalEval(text);
 return text;
 }
 }
});
// Handle cache's special case and global
jQuery.ajaxPrefilter("script", function(s) {
 if (s.cache === undefined) {
 s.cache = false;
 }
 if (s.crossDomain) {
 s.type = "GET";
 }
}
);

```

```

 s.global = false;
 }
});

// Bind script tag hack transport
jQuery.ajaxTransport("script", function(s) {

 // This transport only deals with cross domain requests
 if (s.crossDomain) {

 var script,
 head = document.head || jQuery("head")[0] ||
document.documentElement;

 return {

 send: function(_, callback) {

 script = document.createElement("script");

 script.async = true;

 if (s.scriptCharset) {
 script.charset = s.scriptCharset;
 }

 script.src = s.url;

 // Attach handlers for all browsers
 script.onload = script.onreadystatechange = function(_,
isAbort) {

 if (isAbort || !script.readyState || /loaded|/
complete/.test(script.readyState)) {

 // Handle memory leak in IE
 script.onload = script.onreadystatechange = null;

 // Remove the script
 if (script.parentNode) {
 script.parentNode.removeChild(script);
 }

 // Dereference the script
 script = null;
 }
 }
 }
 }
 }
});

```

```

 // Callback if not abort
 if (!isAbort) {
 callback(200, "success");
 }
 };

 // Circumvent IE6 bugs with base elements (#2709 and #4378)
by prepending
 // Use native DOM manipulation to avoid our domManip AJAX
trickery
 head.insertBefore(script, head.firstChild);
},
abort: function() {
 if (script) {
 script.onload(undefined, true);
 }
};

});

var oldCallbacks = [],
rjsonp = /(=)\?(?=(&|$)|\?)/;

// Default jsonp settings
jQuery.ajaxSetup({
 jsonp: "callback",
 jsonpCallback: function() {
 var callback = oldCallbacks.pop() || (jQuery.expando + "_" + (ajax_nonce+
+));
 this[callback] = true;
 return callback;
 }
});

// Detect, normalize options and install callbacks for jsonp requests
jQuery.ajaxPrefilter("json jsonp", function(s, originalSettings, jqXHR) {

 var callbackName, overwritten, responseContainer,
 jsonProp = s.jsonp !== false && (rjsonp.test(s.url) ?
 "url" :
 typeof s.data === "string" && !(s.contentType ||
 "").indexOf("application/x-www-form-urlencoded") && rjsonp.test(s.data) && "data"

```

```

);

// Handle iff the expected data type is "jsonp" or we have a parameter to set
if (jsonProp || s.dataTypes[0] === "jsonp") {

 // Get callback name, remembering preexisting value associated with it
 callbackName = s.jsonpCallback = jQueryisFunction(s.jsonpCallback) ?
 s.jsonpCallback() :
 s.jsonpCallback;

 // Insert callback into url or form data
 if (jsonProp) {
 s[jsonProp] = s[jsonProp].replace(rjsonp, "$1" + callbackName);
 } else if (s.jsonp !== false) {
 s.url += (ajax_rquery.test(s.url) ? "&" : "?") + s.jsonp + "=" +
callbackName;
 }

 // Use data converter to retrieve json after script execution
 s.converters["script json"] = function() {
 if (!responseContainer) {
 jQuery.error(callbackName + " was not called");
 }
 return responseContainer[0];
 };

 // force json dataType
 s.dataTypes[0] = "json";

 // Install callback
 overwritten = window[callbackName];
 window[callbackName] = function() {
 responseContainer = arguments;
 };

 // Clean-up function (fires after converters)
 jqXHR.always(function() {
 // Restore preexisting value
 window[callbackName] = overwritten;

 // Save back as free
 if (s[callbackName]) {
 // make sure that re-using the options doesn't screw things
around
 s.jsonpCallback = originalSettings.jsonpCallback;
 }
 });
}

```

```

 // save the callback name for future use
 oldCallbacks.push(callbackName);
 }

 // Call if it was a function and we have a response
 if (responseContainer && jQueryisFunction(overwritten)) {
 overwritten(responseContainer[0]);
 }

 responseContainer = overwritten = undefined;
});

// Delegate to script
return "script";
}

});

var xhrCallbacks, xhrSupported,
 xhrId = 0,
 // #5280: Internet Explorer will keep connections alive if we don't abort on unload
 xhrOnUnloadAbort = window.ActiveXObject && function() {
 // Abort all pending requests
 var key;
 for (key in xhrCallbacks) {
 xhrCallbacks[key](undefined, true);
 }
 };
}

// Functions to create xhers
function createStandardXHR() {
 try {
 return new window.XMLHttpRequest();
 } catch(e) {}
}

function createActiveXHR() {
 try {
 return new window.ActiveXObject("Microsoft.XMLHTTP");
 } catch(e) {}
}

// Create the request object
// (This is still attached to ajaxSettings for backward compatibility)
jQuery.ajaxSettings.xhr = window.ActiveXObject ?
 /* Microsoft failed to properly

```

```

* implement the XMLHttpRequest in IE7 (can't request local files),
* so we use the ActiveXObject when it is available
* Additionally XMLHttpRequest can be disabled in IE7/IE8 so
* we need a fallback.
*/
function() {
 return !this.isLocal && createStandardXHR() || createActiveXHR();
} :
// For all other browsers, use the standard XMLHttpRequest object
createStandardXHR;

// Determine support properties
xhrSupported = jQuery.ajaxSettings.xhr();
jQuery.support.cors = !!xhrSupported && ("withCredentials" in xhrSupported);
xhrSupported = jQuery.support.ajax = !!xhrSupported;

// Create transport if the browser can provide an xhr
if (xhrSupported) {

 jQuery.ajaxTransport(function(s) {
 // Cross domain only allowed if supported through XMLHttpRequest
 if (!s.crossDomain || jQuery.support.cors) {

 var callback;

 return {
 send: function(headers, complete) {

 // Get a new xhr
 var handle, i,
 xhr = s.xhr();

 // Open the socket
 // Passing null username, generates a login popup on
Opera (#2865)
 if (s.username) {
 xhr.open(s.type, s.url, s.async, s.username,
s.password);
 } else {
 xhr.open(s.type, s.url, s.async);
 }

 // Apply custom fields if provided
 if (s.xhrFields) {
 for (i in s.xhrFields) {

```

```

 xhr[i] = s.xhrFields[i];
 }
}

// Override mime type if needed
if (s.mimeType && xhr.overrideMimeType) {
 xhr.overrideMimeType(s.mimeType);
}

// X-Requested-With header
// For cross-domain requests, seeing as conditions for a
preflight are
sure.

// akin to a jigsaw puzzle, we simply never set it to be
using ajaxSetup)
// (it can always be set on a per-request basis or even
already provided.
// For same-domain requests, won't change header if
if (!s.crossDomain && !headers["X-Requested-With"]) {
 headers["X-Requested-With"] =
"XMLHttpRequest";
}

// Need an extra try/catch for cross domain requests in
Firefox 3
try {
 for (i in headers) {
 xhr.setRequestHeader(i, headers[i]);
 }
} catch(err) {}

// Do send the request
// This may raise an exception which is actually
// handled in jQuery.ajax (so no try/catch here)
xhr.send((s.hasContent && s.data) || null);

// Listener
callback = function(_, isAbort) {
 var status, responseHeaders, statusText,
responses;

 // Firefox throws exceptions when accessing
properties
 // of an xhr when a network error occurred

```

```

// http://helpful.knobs-dials.com/index.php/
Component_returned_failure_code:_0x80040111_(NS_ERROR_NOT_AVAILABLE)
try {

 // Was never called and is aborted or
 complete
 if (callback && (isAbort || xhr.readyState
==== 4)) {

 // Only called once
 callback = undefined;

 // Do not keep as active anymore
 if (handle) {
 xhr.onreadystatechange =
jQuery.noop;
 if (xhrOnUnloadAbort) {
 delete
 }
 }

 // If it's an abort
 if (isAbort) {
 // Abort it manually if
 needed
 if (xhr.readyState !== 4) {
 xhr.abort();
 }
 } else {
 responses = {};
 status = xhr.status;
 responseHeaders =
 xhr.getAllResponseHeaders();

 // When requesting binary
 data, IE6-9 will throw an exception
 // on any attempt to access
 responseText (#11426)
 if (typeof xhr.responseText
==== "string") {
 responses.text =
 xhr.responseText;
 }
 }
 }
}

```



```

};

if (!s.async) {
 // if we're in sync mode we fire the callback
 callback();
} else if (xhr.readyState === 4) {
 // (IE6 & IE7) if it's in cache and has been
 // retrieved directly we need to fire the callback
 setTimeout(callback);
} else {
 handle = ++xhrId;
 if (xhrOnUnloadAbort) {
 // Create the active xhrs callbacks list if
needed
 // and attach the unload handler
 if (!xhrCallbacks) {
 xhrCallbacks = {};
 }

jQuery(window).unload(xhrOnUnloadAbort);
 }
 // Add to list of active xhrs callbacks
 xhrCallbacks[handle] = callback;
 }
 xhr.onreadystatechange = callback;
}
},
abort: function() {
 if (callback) {
 callback(undefined, true);
 }
},
});
});
}

var fxNow, timerId,
 rfxtypes = /^(?:toggle|show|hide)$/,
 rfxnum = new RegExp("^(?:([+-])=)([" + core_pnum + ")])([a-z%]*$)", "i"),
 rrun = /queueHooks$/,
 animationPrefilters = [defaultPrefilter],
 tweeners = {
 "*": [function(prop, value) {
 var tween = this.createTween(prop, value),
 target = tween.cur(),

```

```

parts = rfxnum.exec(value),
unit = parts && parts[3] || (jQuery.cssNumber[prop] ? "" :
"px"),

// Starting value computation is required for potential unit
mismatches
start = (jQuery.cssNumber[prop] || unit !== "px" && +target)
&&
rfxnum.exec(jQuery.css(tween.elem, prop)),
scale = 1,
maxIterations = 20;

if (start && start[3] !== unit) {
 // Trust units reported by jQuery.css
 unit = unit || start[3];

 // Make sure we update the tween properties later on
 parts = parts || [];

 // Iteratively approximate from a nonzero starting point
 start = +target || 1;

 do {
 // If previous iteration zeroed out, double until we get
something
 // Use a string for doubling factor so we don't
accidentally see scale as unchanged below
 scale = scale || ".5";

 // Adjust and apply
 start = start / scale;
 jQuery.style(tween.elem, prop, start + unit);

 // Update scale, tolerating zero or NaN from tween.cur()
 // And breaking the loop if scale is unchanged or perfect, or if
we've just had enough
 } while (scale !== (scale = tween.cur() / target) && scale !== 1
&& --maxIterations);
}

// Update tween properties
if (parts) {
 start = tween.start = +start || +target || 0;
 tween.unit = unit;
}

```

```

// If a +=/-= token was provided, we're doing a relative
animation
 tween.end = parts[1] ?
 start + (parts[1] + 1) * parts[2] :
 +parts[2];
 }

 return tween;
}
};

// Animations created synchronously will run synchronously
function createFxNow() {
 setTimeout(function() {
 fxNow = undefined;
 });
 return (fxNow = jQuery.now());
}

function createTween(value, prop, animation) {
 var tween,
 collection = (tweener[prop] || []).concat(tweener["*"]),
 index = 0,
 length = collection.length;
 for (; index < length; index++) {
 if ((tween = collection[index].call(animation, prop, value))) {

 // we're done with this property
 return tween;
 }
 }
}

function Animation(elem, properties, options) {
 var result,
 stopped,
 index = 0,
 length = animationPrefilters.length,
 deferred = jQuery.Deferred().always(function() {
 // don't match elem in the :animated selector
 delete tick.elem;
 }),
 tick = function() {
 if (stopped) {
 return false;
 }
 }
};

```

```

 }

 var currentTime = fxNow || createFxNow(),
 remaining = Math.max(0, animation.startTime +
animation.duration - currentTime),
 // archaic crash bug won't allow us to use 1 - (0.5 || 0)
(#12497)
 temp = remaining / animation.duration || 0,
 percent = 1 - temp,
 index = 0,
 length = animation.tweens.length;

 for (; index < length ; index++) {
 animation.tweens[index].run(percent);
 }

 deferred.notifyWith(elem, [animation, percent, remaining]);

 if (percent < 1 && length) {
 return remaining;
 } else {
 deferred.resolveWith(elem, [animation]);
 return false;
 }
 },
 animation = deferred.promise({
 elem: elem,
 props: jQuery.extend({}, properties),
 opts: jQuery.extend(true, { specialEasing: {} }, options),
 originalProperties: properties,
 originalOptions: options,
 startTime: fxNow || createFxNow(),
 duration: options.duration,
 tweens: [],
 createTween: function(prop, end) {
 var tween = jQuery.Tween(elem, animation.opts, prop, end,
 animation.opts.specialEasing[prop] ||
animation.opts.easing);
 animation.tweens.push(tween);
 return tween;
 },
 stop: function(gotoEnd) {
 var index = 0,
 // if we are going to the end, we want to run all the
tweens
 // otherwise we skip this part

```

```

 length = gotoEnd ? animation.tweens.length : 0;
 if (stopped) {
 return this;
 }
 stopped = true;
 for (; index < length ; index++) {
 animation.tweens[index].run(1);
 }

 // resolve when we played the last frame
 // otherwise, reject
 if (gotoEnd) {
 deferred.resolveWith(elem, [animation, gotoEnd]);
 } else {
 deferred.rejectWith(elem, [animation, gotoEnd]);
 }
 return this;
 }
}),
props = animation.props;

propFilter(props, animation.opts.specialEasing);

for (; index < length ; index++) {
 result = animationPrefilters[index].call(animation, elem, props,
animation.opts);
 if (result) {
 return result;
 }
}

jQuery.map(props, createTween, animation);

if (jQuery.isFunction(animation.opts.start)) {
 animation.opts.start.call(elem, animation);
}

jQuery.fx.timer(
 jQuery.extend(tick, {
 elem: elem,
 anim: animation,
 queue: animation.opts.queue
 })
);

```

```

// attach callbacks from options
return animation.progress(animation.opts.progress)
 .done(animation.opts.done, animation.opts.complete)
 .fail(animation.opts.fail)
 .always(animation.opts.always);
}

function propFilter(props, specialEasing) {
 var index, name, easing, value, hooks;

 // camelCase, specialEasing and expand cssHook pass
 for (index in props) {
 name = jQuery.camelCase(index);
 easing = specialEasing[name];
 value = props[index];
 if (jQuery.isArray(value)) {
 easing = value[1];
 value = props[index] = value[0];
 }

 if (index !== name) {
 props[name] = value;
 delete props[index];
 }

 hooks = jQuery.cssHooks[name];
 if (hooks && "expand" in hooks) {
 value = hooks.expand(value);
 delete props[name];

 // not quite $.extend, this wont overwrite keys already present.
 // also - reusing 'index' from above because we have the correct
 "name"
 for (index in value) {
 if (!(index in props)) {
 props[index] = value[index];
 specialEasing[index] = easing;
 }
 }
 } else {
 specialEasing[name] = easing;
 }
 }
}

```

```

jQuery.Animation = jQuery.extend(Animation, {

 tweener: function(props, callback) {
 if (jQuery.isFunction(props)) {
 callback = props;
 props = ["*"];
 } else {
 props = props.split(" ");
 }

 var prop,
 index = 0,
 length = props.length;

 for (; index < length ; index++) {
 prop = props[index];
 tweeners[prop] = tweeners[prop] || [];
 tweeners[prop].unshift(callback);
 }
 },

 prefilter: function(callback, prepend) {
 if (prepend) {
 animationPrefilters.unshift(callback);
 } else {
 animationPrefilters.push(callback);
 }
 }
});

function defaultPrefilter(elem, props, opts) {
 /* jshint validthis: true */
 var prop, value, toggle, tween, hooks, oldfire,
 anim = this,
 orig = {},
 style = elem.style,
 hidden = elem.nodeType && isHidden(elem),
 dataShow = jQuery._data(elem, "fxshow");

 // handle queue: false promises
 if (!opts.queue) {
 hooks = jQuery._queueHooks(elem, "fx");
 if (hooks.unqueued == null) {
 hooks.unqueued = 0;
 oldfire = hooks.empty.fire;

```

```

 hooks.empty.fire = function() {
 if (!hooks.unqueued) {
 oldfire();
 }
 };
 }
 hooks.unqueued++;

anim.always(function() {
 // doing this makes sure that the complete handler will be called
 // before this completes
 anim.always(function() {
 hooks.unqueued--;
 if (!jQuery.queue(elem, "fx").length) {
 hooks.empty.fire();
 }
 });
});
}

// height/width overflow pass
if (elem.nodeType === 1 && ("height" in props || "width" in props)) {
 // Make sure that nothing sneaks out
 // Record all 3 overflow attributes because IE does not
 // change the overflow attribute when overflowX and
 // overflowY are set to the same value
 opts.overflow = [style.overflow, style.overflowX, style.overflowY];

 // Set display property to inline-block for height/width
 // animations on inline elements that are having width/height animated
 if (jQuery.css(elem, "display") === "inline" &&
 jQuery.css(elem, "float") === "none") {

 // inline-level elements accept inline-block;
 // block-level elements need to be inline with layout
 if (!jQuery.support.inlineBlockNeedsLayout ||
css_defaultDisplay(elem.nodeName) === "inline") {
 style.display = "inline-block";

 } else {
 style.zoom = 1;
 }
 }
}

```

```

if (opts.overflow) {
 style.overflow = "hidden";
 if (!jQuery.support.shrinkWrapBlocks) {
 anim.always(function() {
 style.overflow = opts.overflow[0];
 style.overflowX = opts.overflow[1];
 style.overflowY = opts.overflow[2];
 });
 }
}

// show/hide pass
for (prop in props) {
 value = props[prop];
 if (rfxtypes.exec(value)) {
 delete props[prop];
 toggle = toggle || value === "toggle";
 if (value === (hidden ? "hide" : "show")) {
 continue;
 }
 orig[prop] = dataShow && dataShow[prop] || jQuery.style(elem,
prop);
 }
}

if (!jQuery.isEmptyObject(orig)) {
 if (dataShow) {
 if ("hidden" in dataShow) {
 hidden = dataShow.hidden;
 }
 } else {
 dataShow = jQuery._data(elem, "fxshow", {});
 }

 // store state if its toggle - enables .stop().toggle() to "reverse"
 if (toggle) {
 dataShow.hidden = !hidden;
 }
 if (hidden) {
 jQuery(elem).show();
 } else {
 anim.done(function() {
 jQuery(elem).hide();
 });
 }
}

```

```

 }
 anim.done(function() {
 var prop;
 jQuery._removeData(elem, "fxshow");
 for (prop in orig) {
 jQuery.style(elem, prop, orig[prop]);
 }
 });
 for (prop in orig) {
 tween = createTween(hidden ? dataShow[prop] : 0, prop, anim);

 if (!(prop in dataShow)) {
 dataShow[prop] = tween.start;
 if (hidden) {
 tween.end = tween.start;
 tween.start = prop === "width" || prop === "height" ?
1 : 0;
 }
 }
 }
 }

function Tween(elem, options, prop, end, easing) {
 return new Tween.prototype.init(elem, options, prop, end, easing);
}
jQuery.Tween = Tween;

Tween.prototype = {
 constructor: Tween,
 init: function(elem, options, prop, end, easing, unit) {
 this.elem = elem;
 this.prop = prop;
 this.easing = easing || "swing";
 this.options = options;
 this.start = this.now = this.cur();
 this.end = end;
 this.unit = unit || (jQuery.cssNumber[prop] ? "" : "px");
 },
 cur: function() {
 var hooks = Tween.propHooks[this.prop];

 return hooks && hooks.get ?
hooks.get(this) :
Tween.propHooks._default.get(this);
 }
}

```

```

},
run: function(percent) {
 var eased,
 hooks = Tween.propHooks[this.prop];

 if (this.options.duration) {
 this.pos = eased = jQuery.easing[this.easing](
 percent, this.options.duration * percent, 0, 1,
 this.options.duration
);
 } else {
 this.pos = eased = percent;
 }
 this.now = (this.end - this.start) * eased + this.start;

 if (this.options.step) {
 this.options.step.call(this.elem, this.now, this);
 }

 if (hooks && hooks.set) {
 hooks.set(this);
 } else {
 Tween.propHooks._default.set(this);
 }
 return this;
}
};

Tween.prototype.init.prototype = Tween.prototype;

Tween.propHooks = {
 _default: {
 get: function(tween) {
 var result;

 if (tween.elem[tween.prop] != null &&
 (!tween.elem.style || tween.elem.style[tween.prop] == null)) {
 return tween.elem[tween.prop];
 }

 // passing an empty string as a 3rd parameter to .css will
automatically
 // attempt a parseFloat and fallback to a string if the parse fails
 // so, simple values such as "10px" are parsed to Float.
 // complex values such as "rotate(1rad)" are returned as is.
 }
 }
};

```

```

 result = jQuery.css(tween.elem, tween.prop, "");
 // Empty strings, null, undefined and "auto" are converted to 0.
 return !result || result === "auto" ? 0 : result;
 },
 set: function(tween) {
 // use step hook for back compat - use cssHook if its there - use .style
if its
 // available and use plain properties where available
 if (jQuery.fx.step[tween.prop]) {
 jQuery.fx.step[tween.prop](tween);
 } else if (tween.elem.style &&
(tween.elem.style[jQuery.cssProps[tween.prop]] != null || jQuery.cssHooks[tween.prop])
) {
 jQuery.style(tween.elem, tween.prop, tween.now + tween.unit
);
 } else {
 tween.elem[tween.prop] = tween.now;
 }
 }
};

// Support: IE <=9
// Panic based approach to setting things on disconnected nodes

Tween.propHooks.scrollTop = Tween.propHooks.scrollLeft = {
 set: function(tween) {
 if (tween.elem.nodeType && tween.elem.parentNode) {
 tween.elem[tween.prop] = tween.now;
 }
 }
};

jQuery.each(["toggle", "show", "hide"], function(i, name) {
 var cssFn = jQuery.fn[name];
 jQuery.fn[name] = function(speed, easing, callback) {
 return speed == null || typeof speed === "boolean" ?
 cssFn.apply(this, arguments) :
 this.animate(genFx(name, true), speed, easing, callback);
 };
});

jQuery.fn.extend({
 fadeTo: function(speed, to, easing, callback) {

```

```

// show any hidden elements after setting opacity to 0
return this.filter(.isHidden).css("opacity", 0).show()

 // animate to the value specified
 .end().animate({ opacity: to }, speed, easing, callback);
},

animate: function(prop, speed, easing, callback) {
 var empty = jQuery.isEmptyObject(prop),
 optall = jQuery.speed(speed, easing, callback),
 doAnimation = function() {
 // Operate on a copy of prop so per-property easing won't be
lost
 var anim = Animation(this, jQuery.extend({}, prop), optall);

 // Empty animations, or finishing resolves immediately
 if (empty || jQuery._data(this, "finish")) {
 anim.stop(true);
 }
 };
 doAnimation.finish = doAnimation;

 return empty || optall.queue === false ?
 this.each(doAnimation) :
 this.queue(optall.queue, doAnimation);
},
stop: function(type, clearQueue, gotoEnd) {
 var stopQueue = function(hooks) {
 var stop = hooks.stop;
 delete hooks.stop;
 stop(gotoEnd);
 };

 if (typeof type !== "string") {
 gotoEnd = clearQueue;
 clearQueue = type;
 type = undefined;
 }
 if (clearQueue && type !== false) {
 this.queue(type || "fx", []);
 }
}

return this.each(function() {
 var dequeue = true,
 index = type != null && type + "queueHooks",
 timers = jQuery.timers,

```

```

data = jQuery._data(this);

if (index) {
 if (data[index] && data[index].stop) {
 stopQueue(data[index]);
 }
} else {
 for (index in data) {
 if (data[index] && data[index].stop &&
rrun.test(index)) {
 stopQueue(data[index]);
 }
 }
}

for (index = timers.length; index--;) {
 if (timers[index].elem === this && (type == null || timers[index].queue === type)) {
 timers[index].anim.stop(gotoEnd);
 dequeue = false;
 timers.splice(index, 1);
 }
}

// start the next in the queue if the last step wasn't forced
// timers currently will call their complete callbacks, which will
dequeue
// but only if they were gotoEnd
if (dequeue || !gotoEnd) {
 jQuery.dequeue(this, type);
}
});

},
finish: function(type) {
 if (type !== false) {
 type = type || "fx";
 }
 return this.each(function() {
 var index,
 data = jQuery._data(this),
 queue = data[type + "queue"],
 hooks = data[type + "queueHooks"],
 timers = jQuery.timers,
 length = queue ? queue.length : 0;
 });
}
);

```

```

// enable finishing flag on private data
data.finish = true;

// empty the queue first
jQuery.queue(this, type, []);

if (hooks && hooks.stop) {
 hooks.stop.call(this, true);
}

// look for any active animations, and finish them
for (index = timers.length; index--;) {
 if (timers[index].elem === this && timers[index].queue ===
type) {
 timers[index].anim.stop(true);
 timers.splice(index, 1);
 }
}

// look for any animations in the old queue and finish them
for (index = 0; index < length; index++) {
 if (queue[index] && queue[index].finish) {
 queue[index].finish.call(this);
 }
}

// turn off finishing flag
delete data.finish;
};

}

// Generate parameters to create a standard animation
function genFx(type, includeWidth) {
 var which,
 attrs = { height: type },
 i = 0;

 // if we include width, step value is 1 to do all cssExpand values,
 // if we don't include width, step value is 2 to skip over Left and Right
 includeWidth = includeWidth? 1 : 0;
 for(; i < 4 ; i += 2 - includeWidth) {
 which = cssExpand[i];
 attrs["margin" + which] = attrs["padding" + which] = type;
 }
}

```

```

if (includeWidth) {
 attrs.opacity = attrs.width = type;
}

return attrs;
}

// Generate shortcuts for custom animations
jQuery.each({
 slideDown: genFx("show"),
 slideUp: genFx("hide"),
 slideToggle: genFx("toggle"),
 fadeIn: { opacity: "show" },
 fadeOut: { opacity: "hide" },
 fadeToggle: { opacity: "toggle" }
}, function(name, props) {
 jQuery.fn[name] = function(speed, easing, callback) {
 return this.animate(props, speed, easing, callback);
 };
});

jQuery.speed = function(speed, easing, fn) {
 var opt = speed && typeof speed === "object" ? jQuery.extend({}, speed) : {
 complete: fn || !fn && easing ||
 jQueryisFunction(speed) && speed,
 duration: speed,
 easing: fn && easing || easing && !jQueryisFunction(easing) && easing
 };

 opt.duration = jQuery.fx.off ? 0 : typeof opt.duration === "number" ? opt.duration
 :
 opt.duration in jQuery.fx.speeds ? jQuery.fx.speeds[opt.duration] :
 jQuery.fx.speeds._default;

 // normalize opt.queue - true/undefined/null -> "fx"
 if (opt.queue == null || opt.queue === true) {
 opt.queue = "fx";
 }

 // Queueing
 opt.old = opt.complete;

 opt.complete = function() {
 if (jQueryisFunction(opt.old)) {

```

```

 opt.old.call(this);
 }

 if (opt.queue) {
 jQuery.dequeue(this, opt.queue);
 }
};

return opt;
};

jQuery.easing = {
 linear: function(p) {
 return p;
 },
 swing: function(p) {
 return 0.5 - Math.cos(p*Math.PI) / 2;
 }
};

jQuery.timers = [];
jQuery.fx = Tween.prototype.init;
jQuery.fx.tick = function() {
 var timer,
 timers = jQuery.timers,
 i = 0;

 fxNow = jQuery.now();

 for (; i < timers.length; i++) {
 timer = timers[i];
 // Checks the timer has not already been removed
 if (!timer() && timers[i] === timer) {
 timers.splice(i--, 1);
 }
 }

 if (!timers.length) {
 jQuery.fx.stop();
 }
 fxNow = undefined;
};

jQuery.fx.timer = function(timer) {
 if (timer() && jQuery.timers.push(timer)) {

```

```

 jQuery.fx.start();
 }

};

jQuery.fx.interval = 13;

jQuery.fx.start = function() {
 if (!timerId) {
 timerId = setInterval(jQuery.fx.tick, jQuery.fx.interval);
 }
};

jQuery.fx.stop = function() {
 clearInterval(timerId);
 timerId = null;
};

jQuery.fx.speeds = {
 slow: 600,
 fast: 200,
 // Default speed
 _default: 400
};

// Back Compat <1.8 extension point
jQuery.fx.step = {};

if (jQuery.expr && jQuery.expr.filters) {
 jQuery.expr.filters.animated = function(elem) {
 return jQuery.grep(jQuery.timers, function(fn) {
 return elem === fn.elem;
 }).length;
 };
}

jQuery.fn.offset = function(options) {
 if (arguments.length) {
 return options === undefined ?
 this :
 this.each(function(i) {
 jQuery.offset.setOffset(this, options, i);
 });
 }

 var docElem, win,
 box = { top: 0, left: 0 },

```

```

elem = this[0],
doc = elem && elem.ownerDocument;

if (!doc) {
 return;
}

docElem = doc.documentElement;

// Make sure it's not a disconnected DOM node
if (!jQuery.contains(docElem, elem)) {
 return box;
}

// If we don't have gBCR, just use 0,0 rather than error
// BlackBerry 5, iOS 3 (original iPhone)
if (typeof elem.getBoundingClientRect !== core_undefined) {
 box = elem.getBoundingClientRect();
}
win = getWindow(doc);
return {
 top: box.top + (win.pageYOffset || docElem.scrollTop) -
(docElem.clientTop || 0),
 left: box.left + (win.pageXOffset || docElem.scrollLeft) - (docElem.clientLeft
|| 0)
};
};

jQuery.offset = {

setOffset: function(elem, options, i) {
 var position = jQuery.css(elem, "position");

 // set position first, in-case top/left are set even on static elem
 if (position === "static") {
 elem.style.position = "relative";
 }

 var curElem = jQuery(elem),
 curOffset = curElem.offset(),
 curCSSTop = jQuery.css(elem, "top"),
 curCSSLeft = jQuery.css(elem, "left"),
 calculatePosition = (position === "absolute" || position ===
"fixed") && jQuery.inArray("auto", [curCSSTop, curCSSLeft]) > -1,
 props = {}, curPosition = {}, curTop, curLeft;
}
}

```

```
// need to be able to calculate position if either top or left is auto and
position is either absolute or fixed
if (calculatePosition) {
 curPosition = curElem.position();
 curTop = curPosition.top;
 curLeft = curPosition.left;
} else {
 curTop = parseFloat(curCSSTop) || 0;
 curLeft = parseFloat(curCSSLeft) || 0;
}

if (jQuery.isFunction(options)) {
 options = options.call(elem, i, curOffset);
}

if (options.top != null) {
 props.top = (options.top - curOffset.top) + curTop;
}
if (options.left != null) {
 props.left = (options.left - curOffset.left) + curLeft;
}

if ("using" in options) {
 options.using.call(elem, props);
} else {
 curElem.css(props);
}
};
```

```
jQuery.fn.extend({

 position: function() {
 if (!this[0]) {
 return;
 }

 var offsetParent, offset,
 parentOffset = { top: 0, left: 0 },
 elem = this[0];

 // fixed elements are offset from window (parentOffset = {top:0, left: 0},
 because it is it's only offset parent
```

```

if (jQuery.css(elem, "position") === "fixed") {
 // we assume that getBoundingClientRect is available when
computed position is fixed
 offset = elem.getBoundingClientRect();
} else {
 // Get *real* offsetParent
 offsetParent = this.offsetParent();

 // Get correct offsets
 offset = this.offset();
 if (!jQuery.nodeName(offsetParent[0], "html")) {
 parentOffset = offsetParent.offset();
 }

 // Add offsetParent borders
 parentOffset.top += jQuery.css(offsetParent[0], "borderTopWidth",
true);
 parentOffset.left += jQuery.css(offsetParent[0], "borderLeftWidth",
true);
}

// Subtract parent offsets and element margins
// note: when an element has margin: auto the offsetLeft and marginLeft
// are the same in Safari causing offset.left to incorrectly be 0
return {
 top: offset.top - parentOffset.top - jQuery.css(elem, "marginTop",
true),
 left: offset.left - parentOffset.left - jQuery.css(elem, "marginLeft",
true)
};

offsetParent: function() {
 return this.map(function() {
 var offsetParent = this.offsetParent || docElem;
 while (offsetParent && (!jQuery.nodeName(offsetParent, "html") &&
jQuery.css(offsetParent, "position") === "static")) {
 offsetParent = offsetParent.offsetParent;
 }
 return offsetParent || docElem;
 });
}
);

```

```

// Create scrollLeft and scrollTop methods
jQuery.each({scrollLeft: "pageXOffset", scrollTop: "pageYOffset"}, function(method,
prop) {
 var top = /Y/.test(prop);
 jQuery.fn[method] = function(val) {
 return jQuery.access(this, function(elem, method, val) {
 var win = getWindow(elem);

 if (val === undefined) {
 return win ? (prop in win) ? win[prop] :
 win.document.documentElement[method] :
 elem[method];
 }

 if (win) {
 win.scrollTo(
 !top ? val : jQuery(win).scrollLeft(),
 top ? val : jQuery(win).scrollTop()
);
 }

 } else {
 elem[method] = val;
 }
 }, method, val, arguments.length, null);
 };
});

function getWindow(elem) {
 return jQuery.isWindow(elem) ?
 elem :
 elem.nodeType === 9 ?
 elem.defaultView || elem.parentWindow :
 false;
}

// Create innerHeight, innerWidth, height, width, outerHeight and outerWidth methods
jQuery.each({ Height: "height", Width: "width" }, function(name, type) {
 jQuery.each({ padding: "inner" + name, content: type, ":" : "outer" + name },
 function(defaultExtra, funcName) {
 // margin is only for outerHeight, outerWidth
 jQuery.fn[funcName] = function(margin, value) {
 var chainable = arguments.length && (defaultExtra || typeof margin !
== "boolean"),
 extra = defaultExtra || (margin === true || value === true ?
"margin" : "border");

```

```

return jQuery.access(this, function(elem, type, value) {
 var doc;

 if (jQuery.isWindow(elem)) {
 // As of 5/8/2012 this will yield incorrect results for
Mobile Safari, but there
 // isn't a whole lot we can do. See pull request at this
URL for discussion:
 // https://github.com/jquery/jquery/pull/764
 return elem.documentElement["client" +
name];
 }

 // Get document width or height
 if (elem.nodeType === 9) {
 doc = elem.documentElement;

 // Either scroll[Width/Height] or offset[Width/Height] or
client[Width/Height], whichever is greatest
 // unfortunately, this causes bug #3838 in IE6/8 only, but
there is currently no good, small way to fix it.
 return Math.max(
 elem.body["scroll" + name], doc["scroll" +
name],
 elem.body["offset" + name], doc["offset" +
name],
 doc["client" + name]
);
 }

 return value === undefined ?
 // Get width or height on the element, requesting but
not forcing parseFloat
 jQuery.css(elem, type, extra) :

 // Set width or height on the element
 jQuery.style(elem, type, value, extra);
 }, type, chainable ? margin : undefined, chainable, null);
};

});

// Limit scope pollution from any deprecated API
// (function() {

```

```
// The number of elements contained in the matched element set
jQuery.fn.size = function() {
 return this.length;
};

jQuery.fn.andSelf = jQuery.fn.addBack;

// })();
if (typeof module === "object" && module && typeof module.exports === "object") {
 // Expose jQuery as module.exports in loaders that implement the Node
 // module pattern (including browserify). Do not create the global, since
 // the user will be storing it themselves locally, and globals are frowned
 // upon in the Node module world.
 module.exports = jQuery;
} else {
 // Otherwise expose jQuery to the global object as usual
 window.jQuery = window.$ = jQuery;

 // Register as a named AMD module, since jQuery can be concatenated with other
 // files that may use define, but not via a proper concatenation script that
 // understands anonymous AMD modules. A named AMD is safest and most
 robust

 // way to register. Lowercase jquery is used because AMD module names are
 // derived from file names, and jQuery is normally delivered in a lowercase
 // file name. Do this after creating the global so that if an AMD module wants
 // to call noConflict to hide this version of jQuery, it will work.
 if (typeof define === "function" && define.amd) {
 define("jquery", [], function () { return jQuery; });
 }
}

})(window);
```

# Conclusion

To conclude, this website is where the publications data are going to be published. Each user will have different privileges and can see his own publications or search for his publications. Also for administrator privileges there is an option to view and search and publish all publications.

The menu has 6 options:

1. Books
2. Journals
3. Books Chapters
4. Conferences
5. Other
6. Patents

Additionally after the user enters each of those options, a menu appears with handling options such as add, delete, update or search. For search there is a new page where there is first going to be a text box asking for the name of the publication . If the publication already exists in the system the rest of the fields are going to be added automatically.

Note: During the import settings there must be an option where the user have the option to replace completely overwrite to the latest version of the publication or import a new one.

Very important to say that that all of these publications are going to be added dynamically and you should be able to preview them at tables.

To the system there is going to be only one admin (secretary) but there are going to be several users who can be added dynamically (they have to sign up).

User can:

1. Submit publications
2. Be connected with the system and there should be a menu where publications are going to be added
3. Preview any information about the publications and theses
4. Search an author and find publications
5. Author should be able to get a report with the publications with categories (books table, conferences etc)
6. Search all page ( search publications, author(s), year(s) )
7. Can add/edit/delete/update/import his own publications ( If the User\_ID matches on Log In )
8. Not add/edit/delete/update others publications ( only Admin is allowed)

# Bibliography

1. Anton et al., 1994] A. I. Anton, W.M. McCracken and C. Potts "Goal Decomposition and Scenario Analysis in Business Process Engineering", Advanced Information Systems Engineering, Proc. 6th Int. Conf. CAiSE'94, Utrecht, Springer-Verlag Lecture Notes in Computer Science, 811, 1994.
2. Boehm, 1981] B. Boehm, Software Engineering Economics, Prentice- Hall, Englewood Cliffs, New Jersey, 1981.
3. Davis, 1990] A. Davis, Software Requirements : Analysis & Specification, Prentice-Hall, 1990.
4. Dardenne, 1993] A. Dardenne, "On the Use of Scenarios in Requirements Acquisition", Oregon Technical Report, CIS-TR-93-17, 1993.
5. Humphrey, 1989] W. S. Humphrey, Managing the Software Process, Addison Wesley Publishing, 1989.
6. Hudson and Smith, 1996] S. Hudson and I. Smith, "Ultra-Lightweight Constraints", Proc. of ACM Sympo. On User Interface Software and Technology, pp. 147-155, 1996.
7. Isakowitz et al., 1995] T. Isakowitz, E. A. Stohr, and P. Balasubramanian, "RMM: A Methodology for Structured Hypermedia Design", Comm. ACM, Vol. 38, No. 8, August 1995, pp. 34-44.
8. Jacobson, 1992] Jacobson, I., "Object-Oriented Software Engineering: A Use-case Driven Approach", Addison-Wesley, 1992.
9. Karat and Bennet, 1991] Karat, J. and Bennett, J., "Using Scenarios in Design Meetings ÷ A Case Study Example", in Karat, J. (ed.), Taking Software Design Seriously, Academic Press, 1991.
10. Kumar, 1995] V. Kumar, Mbone: Interactive Multimedia on the Internet, Macmillan Publishing, Simon & Schusternet, 1995.
11. Nielsen, 1995] J. Nielsen, Multimedia and Hypertext the Internet and Beyond, Academic Press, 1995

12. Potts, 1995] C. Potts, "Using Schematic Scenarios to Understand User Needs", Proc. Sympo. Designing Interactive Systems: Processes, Practices, Methods & Techniques (DIS'95), August 23-25, 1995, pp. 247-256.
13. Potts et al., 1994] C. Potts, K. Takahashi and A. I. Anton, "Inquiry-based Scenario Analysis of System Requirements", IEEE Software, ??, March 1994, pp. 21-32.
14. Rubin and Goldberg, 1992] K. S. Rubin and A. Goldberg, "Object Behavior Analysis", "Object Behavior Analysis", Comm. ACM., Vol. 35, No. 9, 1992, pp. 48-62.
15. Sano, 1996] D. Sano, Designing Large-scale Web Sites, Wiley Computer Publishing, 1996.
16. Schwabe et al., 1996] D. Schwabe, G. Rossi, and S. D. J. Barbosa, "Systematic Hypermedia Application Design with OOHDM", Proc. Hypertext '96, 1996, pp. 116 - 128.
17. Rossi, G., Schwabe, D., & Lyardet, F. (2000). User interface patterns for hypermedia application. Proceedings Advanced Visual Interfaces Conference 2000, (136–142).
18. Schranz, M.W. (1998). Engineering flexible World Wide Web services. Proceedings SAC Conference 1998, (712–718).
19. Schewe, K.-D. (2000). UML — a modern dinosaur. Proceedings European-Japanese Conference on Information Modeling and Knowledge Bases, Finland.
20. Schewe, K.-D. & Thalheim, B. (2000). Conceptual development of Internet sites. Full-day tutorial at ER Conference 2000, Salt Lake City. Snodgrass, R.T. (2000). Developing time-oriented database applications in SQL. San Francisco, CA: Morgan Kaufman Publishers.
21. Thalheim, B. (2000). Entity-relationship modeling — Fundamentals of database technology. Heidelberg: Springer-Verlag.
22. Thalheim, B. (2000b). Readings in fundamentals of interaction in information systems. Reprint, BTU-Cottbus, <http://www.informatik.tu-cottbus.de/~thalheim>.
23. Thalheim, B. (2001). ASM specification of Internet information services. Proceedings Eurocast Conference 2001, Las Palmas, (301–304). Wegner, P. & Goldin, F. (1999). Interaction as a Framework for Modeling. Conceptual Modeling: Current Issues and Future Directions, Chen, P.P. et al., (Eds.), LNCS 1565, (243–247), Heidelberg: Springer-Verlag.

24. Gaedke, M. & Turowski, K. (1999). Generic Web-based federation of business application systems for e-commerce applications. Proceedings EFIS Conference 1999, 25–42.
25. Göschka, K. & Schranz, M. (2001). Client and legacy integration in object oriented Web engineering. *IEEE Multimedia*, 8(1), 32–41.
26. Garzotto, F., Paolini, P., & Schwabe, D. (1993). HDM — A model-based approach to hypertext application design. *TOIS*, 1(1), 1–26.
27. Goldin, D., Srinivasa, S., & Thalheim, B. (2000). IS = DBS + interaction - towards principles of information systems. *Proceedings ER Conference 2000, LNCS 1920*, (140–153), Heidelberg: Springer-Verlag.
- Güell, N., Schwabe, D., & Vilain, P. (2000). Modeling interactions and navigation in Web applications. *Proceedings ER Workshops 2000, LNCS 1921*, (115–127), Heidelberg: Springer-Verlag.
- Hennicker, R. & Koch, N. (2000). A UML-based methodology for hypermedia design. *Proceedings UML Conference 2000, LNCS 1939*, (410–424), Heidelberg: Springer-Verlag.
- Horrocks, I. (1999). Constructing the user interface with statecharts. Reading, MA: Addison-Wesley.
- Isakowitz, T., Kamis, A., & Koufaris, M. (1998). The extended RMM methodology for Web publishing. New York University. Available under <http://rrr-java.stern.nyu.edu/rmm/>
- Lewerenz, J. (2000). Human-computer interaction in heterogeneous and dynamic environments: A framework for its conceptual modeling and automatic customization. Ph.D., BTU Cottbus.
- Lewerenz, J., Schewe, K.-D., & Thalheim, B. (1999). Modeling data warehouses and OLAP applications by means of dialogue objects. *Proceedings ER Conference 1999, LNCS 1728*, (354–369), Heidelberg: Springer-Verlag.