Course:
Web Technologies for Artificial Intelligence

Programme: Artificial Intelligence

Course Coordinator: Yuri Demchenko
Contact: y.demchenko@uva.nl

Course teaching assistant: Nick de Wolf
Contact: N.J.G.deWolf@uva.nl

Course guidelines

1. Group project organisation

During the course the students will work in groups to develop fully functional dynamic interactive website that support interaction between user and provider or server. There are several use cases or web applications are proposed in Section 2 below from which the group can select. You make your choice in consultations with the groups on the first day of the first course week.

The groups are pre-assembled. Each member in a group is expected to offer an equal and significant contribution to the development of the project. The division of tasks within a group is always such that everyone should have hands on with all the components of the web application, and not "per item" as in: Alex makes all web pages, Brenda writes all code, Chris creates the database, etc. Each group chooses one contact person who will contact in case of problems the student assistant assigned to the group (in a first instance) or the teacher (in the second instance). Group organises its work independently. For the project, the study load is 42 hours per week.

Each group is assigned a student assistant who will provide necessary guidelines approximately half an hour twice a week. See the consultation schedule on Blackboard. Attendance at this consultation is mandatory for all members of the group. Please prepare questions to your student assistant for these consultations. In particular, limit individual questions by email. The student assistant has three roles: assistant, taskmaster and monitor. As an assistant, he/she can help you find a solution in case of problems. As a taskmaster, the student assistant can answer questions about the web application requirements and functional components. As a monitor, the student assistant keeps track of progress and checks whether each member of the group provides an equal and significant contribution.

Two lectures will take place two times per week and will provide information on the main topics/subjects of this course. In preparation to the lectures you will study material specified for each lecture (see section 1.4: Literature). The rest of the time you will work with your group on the project. This can be done in the designated for this rooms[1] or in other workplaces at the Science Park.

[1] See Datanose. If the reserved rooms are closed at the start of the day, ask at the reception for a key to open them.

Each group has its own Linux web server that has the main software already installed. Use your own laptop to log on to the Linux web server. See section 3 for information about using this server.

In addition to the website itself, you will need to provide a final report at the end of course (week 4) explaining the design and operation of the web application. Section 3 explains what should be delivered.

At the end of the fourth week, all groups will demonstrate their projects to each other and to other invitees during a project demo and defense session. During this demonstration, you will need to explain your design solutions to others and to assessors. Some listeners may ask more critical questions and will eventually be asked which website may be named the Best Website this year.

Additional information and instruction will be provided during the lectures and via Blackboard.

2. Timetable and scheduling

Planning is essential for a project of this type and limited time available. Do not underestimate this! The following schedule is a guideline. The actual schedule will vary by group. Make arrangements within your group! Please take into account the conditions described above to the task allocation. Within PAV, the preparation of a project wwwplan is further dealt with in the first and second weeks.

Table 1. Recommended timeline.

Week	Subjects	Techniques	Result
1	Development of the website, structure and datamodel.	HTML/CSS, Javascript, pen en papier	Prototype structure and visual design, description of data to be used and exchanged.
2	Database development, input via forms, output via database views, SQL queries.	MySQL, phpMyAdmin and PHP	Finish website design, start adding data and usage experience. Backend database is operational and support web application.
3	User facing application part: login and user friendly interface	PHP, Javascript, AJAX	User facing interface and services the website, debugging and fixing.
4	Security, documentation, demonstration.	Final testing and writing report	Web application is ready and fully tested, report finished, demonstration

3. Web server

Each group gets its own Linux web server on which the website should be created. To find the data from this server you must first log in to the course website http://webai.science.uva.nl/. Use your **UvANetID** with its associated password.

Then navigate to the "Server" page. On this page you will find information about your server:

- URL (also called "webadress") of your server: `http://agileXYZ.science.uva.nl`[2]
- Address at which can logging in the webserver; This can be done in two ways:
  - with "IP address": where address is in a form "146.50.38.XYZ", or
  - with so called "fully qualified domain name" (FQDN): part of webaddress standing after "`http://`", that is `agileXYZ.science.uva.nl`
- Your personal username and password to log in to the server. That password, you will notice, is longer and more complicated than you may be used to. That's for a reason:

Warning
Your server is connected to the public Internet. It's a matter of waiting until somebody tries to hack your server. A strong password is important to make those attackers not too easy.

3.1. Logging in

Login to the web server is via SSH ("secure shell"). For this you need an SSH client. If you are using Linux or Mac, an SSH client is already present: ssh. For example, download for Windows, PuTTY (specifically: putty.exe - http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html). The transfer of files between your laptop and the web server is also via the SSH protocol. From the command line with Linux or Mac, use the command "scp" or "rsync -e ssh" for Windows pscp.exe (https://the.earth.li/~sgtatham/putty/latest/x86/pscp.exe). There is also software to transport from a GUI file, such as FileZilla (client - https://filezilla-project.org/).

Consult before starting your group how to transfer files to the web server. If you do this negligently, then it's very easy to override your colleagues' work. It is better to use a version control system, such as "git".

3.2. Git

The use of git is recommended as you work as a group on the same product. Most students have practiced during PAV with the use of git and GitLab (https://gitlab-fnwi.uva.nl/). In GitLab is for Each group also created a "GitLab group". Here you can create a repository for the website that you are going to develop. Try to find the best way to use git on the server. For example, you can have a clone in all ***public_html*** directories per user. If you do not yet have access to GitLab, but would like to use git, send an email to the server administrator: webai-server@list.uva.nl

3.3. Directory location

There are two directory locations on the server that you can use when creating the website and intermediates:

[2] Replace "XYZ" in het getal dat staat aangegeven op de "Server" pagina.

1. `/var/www/html`: de inhoud van deze directory komt beschikbaar op de hoofdpagina van je server: http://agileXYZ.science.uva.nl en https://agileXYZ.science.uva.nl (note: "https" in place of "http").

This directory is where your actual website will be located.

2. `public_html` directory in your home directory (`~/public_html`). Op deze plek kan je experimenteren op een persoonlijke plek. Bestanden die je hier plaatst, komen beschikbaar op: http://agileXYZ.science.uva.nl/~<username>

3.4. MySQL and phpMyAdmin

On every server a MySQL database management system (DBMS) is available with which you can manage the database belonging to your website. Here are two ways to communicate with MySQL:

1) the mysql command-line client; it is only accessible if you are logged in via SSH on the web server. With this client you can directly type in commands that are executed by the MySQL server after which the result is displayed. In general, this method is only used by advanced users. The client is started by the following command (4):
(4) Take over everything that comes after "$" (the "prompt").

   $ mysql –u <*username*> –p

   with `<username>` your personal login name. If you are asked for a password, give the corresponding password.

2) phpMyAdmin;
   This is a web interface to the database, an interactive web application, very similar to what you yourself are going to make (with the difference that this web interface is intended for database administrators and not for end users). There are friendly "point and click" methods for creating and modifying tables. If you want something special, you can fall back on entering an SQL query in the appropriate query window. The address of the phpMyAdmin page on your group server is:
   https://agileXYZ.science.uva.nl/phpmyadmin/

Log in with your personal login name and password.

3.5. Using own domain name

The addresses mentioned above for your website are not very attractive. It is possible to add your own domain name to your website. This is optional. Such a domain name will be on your costs and you can register it with a "domain registrar". Next, you have to change a "DNS A record" in your registration information to refer to the external IP address of the server: that IP address can be found on the "Server" page on the course website. In addition, you also need to do the following on the web server:
edit (as root) `/etc/apache2/sites-available/webdb.conf`

- find line with "`ServerAlias jouwdomeinnaam.nl`" and uncomment it; and put there your registered domain name
- restart Apache, the HTTP server:

$ sudo systemctl restart apache2

- If you also want to enable SSL, so that your domain can also be reached via HTTPS, you have to run the command "sudo certbot" and follow the instructions.

More information follows in the lecture.

3.6. Root-rights and responsibility

Everyone has root rights via "sudo" on the web server. It is important to deal with this responsibly. You can disable the server with the wrong modifications and shut out your group members or yourself!

Read for a brief explanation http://brajeshwar.com/2008/linux-root-power/.

3.7. Problems with the webserver

If there is a problem with the web server that you do not like yourself, please contact the server administrator via ***webai-server@list.uva.nl***. Please note: this address can only be used if there are problems with the web server, not for substantive questions about your website. You retain substantive questions for the consultation with the student assistant assigned to your group.

3.8. Example code

During the lectures frequent use is made of examples. These are available through the web server listed under "Sample code" on the course website (use direct link https://agile100.science.uva.nl). For security reasons, this site is protected with a username / password: that is ***webdb / webdb2019***. In the case of PHP examples you will see that not only the results are visible, but also the PHP code itself. To avoid confusion: that is not "normal behavior". We did this to show you what the code is behind the results.
Voorbeeld.

3.9. Recommended software

Most of the software you need to build a website is already on your web server: it is equipped with the standard "LAMP" software to build a dynamic website: Linux as operating system, Apache as web server, MySQL as database management system, and PHP as back-end scripting language.

Of course, your website must work on all browsers. You do well to test this with multiple, different browsers. Next to your standard browser, try a few others, such as Chrome, Firefox, Edge, Internet Explorer, Opera, Safari and do not forget the browser (s) of your smartphone!

If you are traveling for a long time between Science Park and at home and want to be able to work on the train or bus on the road, it can be useful to install a similar environment on your own laptop. Under Linux you can easily install the same software as is installed on the server. If you use a different operating system, take a look at EasyPHP (or the Development Server), WAMP Server, XAMPP or MAMP (for Mac users).

For editing HTML, CSS, PHP and Javascript files, preferably use an editor that supports "syntax highlighting" so that syntax and typing errors are quickly visible. Under Linux: Atom, vim or Emacs. Under Windows: Atom, Visual Studio Code.Notepad ++.

This course is intended to study the architecture of dynamic websites. It is therefore not allowed to build your website using integrated development software like Wordpress, DreamWeaver, FrontPage, Web Studio, Flash / AIR, etc., or any random CMS ("Content Management System"). Also the use of frameworks like Bootstrap, Ionic or javascript libraries like Angular, Dojo, MooTools, Vue.js, etc. is not allowed. The only exception is jQuery: you can use it. The use of software that support site development, such as HTML / CSS editors, phpMyAdmin, debuggers, http://codepen.io, http://jsbin.com, etc. is allowed.

4 Recommended reading

For this course there is no single textbook. Alle literatuur die je nodig hebt staat op het internet. We gaan er vanuit dat iedereen voorafgaand aan de colleges bekend is met de inhoud van de volgende tutorials:

Week	Tutorials
1 (for second lecture)	http://www.w3schools.com/html/ http://www.w3schools.com/css/ http://www.w3schools.com/js/
2	http://www.w3schools.com/sql/ http://www.w3schools.com/php/
3	http://www.w3schools.com/jquery/
4	http://www.w3schools.com/xml/ajax_intro.asp

During the lectures the teacher will discuss the most important concepts, the "best practices" and the "Pitfalls". The lectures are not compulsory, but what has been told there is known and therefore assessed.

A course page has been set up in Blackboard. All necessary documents are available via "Content".

In general, the answers to all questions can be found via Google on the internet. Make use of it. Use code fragments that you find in this way in a responsible way! In particular: it is allowed to use small code fragments (up to 10 consecutive lines) from the internet without acknowledging the source, but always mention the source for large pieces (more than 10 consecutive lines)!

Below is a list of websites that will come in handy:

- W3C (World Wide Web Consortium) is the organisation that develops and maintains we standards. Website contains all related standards for web technologies:
- http://www.w3.org/

HTML (version 5)

- HTML5 Tag Reference

http://www.w3schools.com/tags/

Markup Validation Service, een service die controleert of je HTML documenten correct zijn opgesteld:
http://validator.w3.org/ http://validator.w3.org/unicorn/

CSS (version 3)
- CSS Property Reference:
http://w3schools.com/cssref/default.asp
- CSS Selector Reference:
http://www.w3schools.com/cssref/css_selectors.asp

PHP (version 7.1)
- http://www.php.net/
- http://www.php.net/manual/en/

MySQL (version 5.5)
- http://dev.mysql.com/doc/refman/5.5/en/
In particular from chapter 11 ("Data Types") till chapter 13 ("SQL Statement Syntax").

Javascript
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference
- https://developer.mozilla.org/en/Gecko_DOM_Reference
- https://developer.mozilla.org/en/DOM/element

5. Web applications

Below is a list of websites and applications that may be useful for you to select your project:

5.1. Website devoted to one of UN defined Sustainable Development Goals (SDG)

> ***Note: This is the main topic recommended for website and web application development in this term of course. Addressing such topics will contribute to common understanding of what are global problems on our planet an dhow they can be targeted by joint efforts.***

17 Sustainable Development Goals are defined in the UN document "Transforming Our World - the 2030 Agenda for Sustainable Development". You can select of goals and develop appealing website with elements of dynamic webpages generation, supporting user community, providing up to date information, and interfacing to available monitoring data. See separate document and links:

- 17Goals http://17goals.org/
- The Global Partnership for Sustainable Development Data http://www.data4sdgs.org/
- Data Revolution Group http://www.undatarevolution.org/
- Center for Open Data Enterprise http://opendataenterprise.org/
- UN Sustainable Development Solutions Network (SDSN) http://unsdsn.org/

By developing website devoted to one of 17 SDG you will address the following questions and include the following components in your website:

1) Website explaining what is the problem, challenge, affected groups of people
2) List of relevant information resource

3) Community supporting services such as events agenda, discussion forum, realtime data monitoring, other
4) Access and visualisation of available monitoring data related to SDG topic and related activities and processes; data can be accessed via WebAPI or collected from relevant websites using webscrapping techniques (webcrawling as an example)

SDG websites can use one of described below web based application.

Other possible topics and website templates

Web applications development can use one of the following topics of templates.

5.2. Open agenda

A large company wants to keep a public "agenda" on the web of events that are important for employees, customers and shareholders.

The agenda can be viewed without logging in. By default, the agenda appears for all target groups and for the current week. Users can change this by indicating to which target group (s) they want to be counted. They can also scroll forward or backward in the agenda. Clicks on an announcement will show details (a piece of text about the event).

People who have rights to do so can log in to prepare messages for the agenda. It can also be indicated for which target group (s) the message is intended. There is a small editor who screens all submitted messages and decides on whether or not to make public. So, there are two groups of users, each with its own interface: one for creating messages and one for approving. Users may only delete their own messages. If a message is changed, it must be screened again.

There is also an administrative section that allows users who can access it: add, accept, delete and change messages, accept users, delete, block or change their rights.

5.3. Educational courses database

A faculty offers various courses that each offer a large number of elective courses. Each elective course runs for a whole period within a semester (one semester consists of three periods of 8 weeks, 8 weeks and 4 weeks), either in the morning or in the afternoon. The intention is that students can register online for these courses. After an attempt to register, they are told: (i) succeeded (ii) failed: maximum number of participants has been reached, (iii) failed: conflicting in terms of schedule with a previously chosen course. After enrolling in one or more subjects, a student can view his timetable and also indicate which space is still to be filled in. To prevent misuse, login with name and password is required.

After logging in, first a personal page appears, with hyperlinks to the home pages of the subjects for which you have previously registered (and which are currently running, or, of the current year). From this page the grid can also be made visible and changed. Other personal facilities can be added to taste.

An administrative part gives the study programs the possibility to make, modify or delete elective courses.

5.4. Online ordering and invoices

A company enables regular customers to place orders via the web. In the public part of the site is a catalog with products (with clickable images). Regular customers can log in. They then see an extra link for each product, with which they can put the product in their shopping cart, whereby the number of copies can also be specified. There is a "check out" link. After this, the customer sees the contents of his orders. It is then still possible to remove certain products from the cart. Then you can continue shopping or make the order final.

Invoicing also goes via the web. Each customer can see an overview of previous orders. With each order it is stated whether it has already been delivered and whether the customer has already paid. Administrators of the site have a separate interface where they can view placed orders. They can indicate per order whether this order has been delivered and whether it has been paid. The admin section also gives the company the possibility to add, change or remove products from the web store.

5.5. Online reservation

A company wants a website that allows users to reserve their services online (for example, a table in a restaurant, an appointment with the GP, a room for meetings or a boat to row in). Three groups of users can be distinguished:

1. Administrators: they must be able to add, change or delete new reservation services. Services can be public or only visible on the basis of membership. Requests for membership are accepted by administrators or rejected. Administrators also get a clear overview of the services that are reserved and can add, change or delete reservations.

2. Everyone can view public services (without logging in). You can also register as a user. Registered users can reserve public objects and view previously made reservations and (if the reservation has not already expired) change and delete. Of course, it is important that reservations do not overlap.

3. Members: Only members are not allowed to reserve public services. Users can submit an application to become a member of the website administrator.

Of course, the user receives an email to confirm when registering an account, when making a reservation, a request for membership and it would be nice if the reservation is available to also get a reminder if the moment of the reservation almost arrives.

5.6. Discussion forum

A software company wants to set up a series of web-support forums so that users of their products can help each other, instead of having to call the expensive support department (almost all companies have such forums). They ask you to create the company website that processes the forum software.

There are three types of users of these forums:

- Administrators (people from the software company itself) must be able to create and cancel new forums - each with its own theme. It is possible to choose for each forum whether messages that have been submitted can be placed directly, or first be screened

by an administrator. Administrators can see a list of submitted messages that they still need to approve or reject. After approval the message becomes visible to everyone, within the forum for which it was intended.
- Registered users can read approved messages and can submit messages themselves via a web form and they can reply to previous messages.
- Everyone can read (without login) approved messages. You can also register as a user. A password is then automatically sent, which changes the user to a registered user.

A good forum works with "threads", so that the relationship between messages and answers becomes visible. When someone responds to a previously posted message, a "thread" is created.
When you open a forum you first see only the messages that are the starting points of all threads. Each message contains hyperlinks to the corresponding answers, and each reply contains a hyperlink to the message to which the answer was given.

5.7. Social network

Create a website that allows users to build a social network. Users create an account and thereby create a profile on which they can place information about themselves. A profile can always be changed by the user and is initially public, unless the user indicates that it does not want to. Users can post messages to their profile that are only visible to themselves and their friends. Users can make friends and respond to messages from friends. A request to become a friend must be honored by the receiving party.

Users can also send each other private messages.

There are also pages of companies or other topics that users can join. In principle, users can not see the content until they are friends or members. There must also be an interface for administrators. Administrators can (temporarily) put everything that users can do and also temporarily put users out of service.

6. Reporting

A lot has to be handed in during the last week. This is done via Blackboard by means of a single zip file containing all files. Here is an overview of the files that have to be submitted and explanation:

6.1 The web application

The web application is a subdirectory in the main directory of the zipfile containing all necessary files, such that a working application/wedsite can be obtained on a web server. Take note of the following:

- Do not forget to add the database; structure and content! A database can be written to a text file with phpMyAdmin or the MySQL command line tool "mysqldump".

6.2 Proof of cooperation

Put in the main directory of the zipfile a file with the name "README.txt" containing:

1. the composition of your project group (names and student numbers). This counts as proof that you have worked together;
2. a URL to the working application on the group server within webai.science.uva.nl or, in the case that you have registered your own domain, a URL that redirects to the group server;
3. if necessary: username and password to log in with the highest possible permissions on the working web application;
4. if applicable: source references of code not developed by the group.

The directory in the zipfile is needed to check your code. The URL is needed to test the application on the server where you have tested it yourself.

6.3. Technical report

Note: two documents must be produced during this course:

- Technical report: written and assessed by the entire project group (20% of the final mark). The tutors do not accompany writing this. The requirements for this document are described below.
- Reflection report: written by each student individually and assessed by the tutor (50% of the process figure or 10% of the final mark). The requirements for this document are dealt with by the tutors during the PAV meetings. Note: the grade for this report must be at least a 5 to complete this course!

The technical report explains the design choices made during the realization of the website and explains how the site has been technically realized. A good report contains at least the following:

- Title: name of the website or project.
- Authors: group name, names of the group members including student number.
- Introduction: describe the goal (what is the website for) and result (what exactly should the website do). Or, what are you responsible for as a team.
- Design: an elaboration of the website: which functional requirements have come from the discussions with the client, eg: describe what an unregistered visitor can, describe what can be registered as a registered user, and describe what one can do as an administrator. What does the data model look like that has been distilled from it? Which tables are there and with which relations? Are the tables 3NF; if not: why not? In this case, show a design drawing ("diagram"), eg generated with phpMyAdmin (see tab "Designer")!
- Implementation: How is the code structured? Where do we find the various developed functionality? How is it prevented that the same code is repeated in several places? Which queries are most important for the operation of your website? How is the website made user-friendly? How is the website protected against malicious users?
- Results: mainly show (a few) screenshot (s)!
- Discussion: a reflection on the result. Does all functionality work as planned? What is not and why not? Is the site "maintainable": is the functionality easy to expand? What had you dealt with differently afterwards?
- Source references: if use has been made of code that has not been developed by the group, please state this here.

The example content described above is a guideline. Deviate may, but make sure that the report has a head, torso and a tail with a red thread through it. The layout described above,

in our experience, almost automatically occurs. In addition, a report must of course look well-groomed and there is nothing to be said about spelling and grammar. Documentation on details (such as the operation of individual functions) is a comment in the code, not in the report! Agree how the paperwork is divided. As a guideline, the report is at least 7 but no more than 15 pages long. Anything previously written down in the project plan may be used for the report.

Put a PDF version of the technical report in the main directory of the zip file to be submitted.

7. Assessment

The assessment is done by the lecturer, a student assistant and from the PAV. The final assessment of this course is determined individually by means of the weighted average mark of the following assessments:

1. (60%) The end product: the operation of the website and the design of the associated database and the program code. We pay attention to:
   a. The database: is the data model correct? Are the right types and attributes used for the description of the various entities? Are the tables 3NF?
   b. The code: is the code correct, well structured and well documented?
   c. Design: are the web pages HTML5 compliant? Has care been given to appearance? Is there a correct separation of content and layout? Is the website "responsive", i.e. does it work on all known browsers?
   d. Navigation / interaction: does the website work according to the specification? Is the use of the website sufficiently intuitive?
   e. Security: is there sufficient protection against abuse by malicious users?

2. (20%) You function in a team; we pay attention to entrepreneurship (own initiative, tackling problems, steering the project), organization (division of tasks, planning), work attitude (presence, punctuality, commitment) and cooperation (responsibility, working atmosphere). 10% for the aforementioned individually written evaluation report on the cooperation (treated in PAV and again: the figure must be at least a 5.0 to complete this course!), The other 10% is based on an assessment by TAs**. If it does not go well with the cooperation: tell us!** This is never charged negatively. What counts is whether at the end the work has been well distributed.

3. (20%) The technical report: is the report well structured, clear