



Курсова робота з веб-програмування

Версія 2.0.0

Версія

У процесі навчання в Комп'ютерній Академії ШАГ студенти напряму «Розробка програмного забезпечення», крім іспитів із предметів, також здають курсові з того чи іншого предмету.

Давайте запитємо себе: чому програма навчання має курсові? Є кілька відповідей на це питання. Перша відповідь: студентам важливо попрацювати над великим комплексним проектом, що об'єднує набір технологій. Наприклад, якщо ми говоримо про курсову з веб-програмування, то в рамках проекту можна й потрібно використовувати максимальний спектр інструментів зі світу веб-програмування. Друга відповідь: студентам після закінчення навчання необхідно мати своє портфоліо, і курсові роботи прекрасно допомагають у цьому.

Курсовий проект із веб-програмування обов'язково повинен складатися з двох частин: клієнтської та серверної. Для реалізації клієнтської частини необхідно використовувати HTML/CSS, JavaScript, jQuery/AngularJS/ReactJS тощо. Для реалізації серверної частини потрібно використовувати PHP або ASP.NET MVC. Обов'язково має бути СКБД для зберігання даних.

Вибір теми курсової роботи

Комп'ютерна Академія ШАГ і викладач, який викладає конкретний курс, не зобов'язані видавати студентові тему курсової роботи. Ми можемо запропонувати тему, але не зобов'язані цього робити. Вкрай важливо стимулювати студента вибрати тему самому. Чому?

Якщо людина вибирає проект, цікавий для неї з тієї чи іншої причини, мотивація роботи над проектом буде набагато вище. Якщо обраний проект буде реальним комерційним замовленням, ми отримаємо ідеальний курсовий проект. Безумовно, тема, що запропонована в якості курсової, повинна відповідати тематиці курсового проекту.

Затвердження теми курсової роботи

Тема курсової роботи студента має бути затверджена завідувачем кафедри за один-два місяці до дати захисту.

Студент описує тему роботи та список можливостей курсового проекту в заявці обсягом не більше ніж два аркуші. На першому кроці отриману заявку стверджує викладач, який відповідає за реалізацію курсового проекту. Наприклад, якщо ми говоримо про курсову з веб-програмування, то це має бути викладач, який викладав курс PHP або ASP.NET. На другому кроці тему стверджує завідувач кафедри «Розробка програмного забезпечення» філії. На третьому кроці заявку студента отримує менеджер навчального процесу конкретного напрямку. Заявка оформляється у двох примірниках. Один примірник залишається в студента, другий – у менеджера навчального процесу. Даний механізм може бути реалізований також в електронному вигляді. Наприклад, менеджер навчального процесу зберігає відскановані заявки з підписами.

Основне завдання в процесі затвердження – це переконатися в тому, що пропонується проект за складністю, технологіями та повнотою відповідає курсовій роботі. І наявність підписів викладача й завідувача кафедри показує, що вони проаналізували тему і впевнені в тому, що курсовий проект підходить під озвучені критерії. Якщо філія не має завідувача кафедри, то підпис ставить тільки викладач.

Звертаємо вашу увагу на те, що курсова робота може бути виконана одним студентом або командою студентів. У командному проекті обсяг роботи кожного студента має бути рівним обсягу курсової роботи для одного студента.

Робота над курсовим проектом

Студент виконує курсову роботу вдома, поза стінами Академії. Якщо в студента виникають питання, то в процесі роботи він питає викладача в рамках навчання.

У навчальному плані стаціонару та ЄКО виділено час на консультації та захист курсового проекту. Виділені пари бажано розташувати не на одному тижні, а розкидати на різні тижні, для того, щоб у студента процес підготовки був плавний. Питанням розподілу консультаційних пар і датою захисту займається менеджер навчального процесу.

У навчальному плані для напівстаціонару виділений час витрачається тільки на захист проекту.

Захист курсового проекту

Дату захисту курсового проекту призначає менеджер навчального процесу після консультацій із викладачем і завідувачем кафедри.

На захисті повинні бути присутніми менеджер навчального процесу напряду (по змозі), викладач, який відповідає за курсову роботу студента, завідувач кафедри. Завідувач кафедри приходить на захист із заявками студентів, які зберігалися в менеджера навчального процесу.

Студент на захист готує роздрукований документ, до складу якого входять:

1. Титульний аркуш із назвою теми, назвою курсового проекту, ПІБ студента.
2. Копія заявки, яка залишилася в студента.
3. UML-діаграми проекту. Обов'язковою умовою є наявність діаграми класів, діаграми варіантів використання (use case diagram). Інші діаграми можуть бути прикладені за бажанням.
4. Схема бази даних (якщо проект має базу даних).
5. Вихідний код курсового проекту (якщо студент не готовий віддати вихідний код проекту з будь-якої причини, то ключові шматки коду).

Захист проводиться у форматі презентації. Студент розповідає про свою роботу перед комісією. Схвалюється використання презентацій у PowerPoint та інших форматах. У рамках презентації студент повинен розповісти про свій проект, чому була обрана дана тема, висвітлити наявність конкурентів, обґрунтувати обрані технології та ін. Після розповіді про свою роботу студент повинен продемонструвати роботу проекту на практиці.

Комісія може ставити будь-які питання до проекту в рамках процесу захисту. Якщо комісія захоче поставити запитання до вихідного коду, студент зобов'язаний продемонструвати місце в коді, яке цікавить комісію.

Максимальна тривалість виступу студента – 15 хвилин, рекомендований час для запитань комісії – 15 хвилин. Разом: загальний час роботи з конкретним студентом на захисті – 30 хвилин.

Після виступу всіх студентів комісія влаштовує колегіальне обговорення оцінок (без присутності студентів). Оцінка конкретному студенту виставляється після досягнення колегіальної згоди або більшості. Після того, як усі оцінки виставлено, завідувач кафедри оголошує їх перед студентами відразу після фінальної наради. Якщо в студента виникає питання щодо виставленої оцінки, завідувач кафедри зобов'язаний відповісти студенту індивідуально після оголошення результатів.

Студенти, які не захистили свою курсову роботу або бажають отримати вищу оцінку, відправляються на перездавання. Дату перездавання призначає менеджер навчального процесу після консультацій із викладачами й завідувачем кафедри.

Приклади тем для курсового проекту з веб-програмування

1. Веб-чат.

- *Короткий опис:* студент має реалізувати веб-чат. Чат повинен мати історію повідомлень, список контактів, приватну, групову та масову розсилку, чорний список, зміну станів користувачів. У реалізації веб-чату обов'язково повинні бути користувачі-адміністратори, які можуть забанити користувачів чату за порушення правил спілкування.

2. Тестовий веб-портал.

- *Короткий опис:* проект для проведення веб-тестування. Тестові завдання та список користувачів зберігаються в базі. Також ведеться облік кожного проходження. Портал має підтримувати різні види питань і завантаження картинок до запитань. Портал ділиться на дві частини: редактор тестів та тестова система. У редакторі тестів створюються тести, які зберігаються в базі. Кожен тест складається з набору питань. Питання містить довільну кількість варіантів відповідей. Питання бувають двох видів: вибір одного варіанта з багатьох, вибір декількох варіантів. Кожне питання має свою вагу. До питання можна завантажувати картинку. Тестова система відтворює тести для користувачів. Користувач заходить у систему під своїм логіном і паролем. Він має обмежену кількість спроб для тестування. Результати вносяться в базу.

3. Веб-проект для голосування.

- *Короткий опис:* веб-проект надає можливість провести голосування. Це може бути голосування в конкурсі «Президент навчального закладу» або схожому конкурсі. Користувачі мають можливість проголосувати за свого кандидата. Система складається з двох частин: клієнтська частина для голосування та інтерфейс адміністратора. Користувач системи для голосування реєструється в клієнтській частині. Після реєстрації у користувача з'являється можливість пройти голосування. Один користувач може проголосувати лише один раз в одному голосуванні. Адміністратор системи може створювати різні голосування, додавати кандидатів, видаляти кандидатів, редагувати кандидатів, встановлювати час початку й закінчення голосування тощо.

4. Веб-планувальник завдань.

- *Короткий опис:* веб-планувальник завдань дозволяє вести списки своїх завдань. Завдання може бути кінцевим або мати список підзавдань. Завдання має пріоритет, дату закінчення, опис, хештег, коментар, прикріплений файл. Користувач може додавати, змінювати, видаляти, шукати завдання. Користувач повинен мати можливість подивитися список завдань на день/тиждень/місяць/рік.

5. Веб-система для створення найпростіших персональних сторінок.

Обов'язкові дві частини: клієнтська та інтерфейс адміністратора.

6. Веб-система здачі будинків в оренду.

Обов'язкові дві частини: клієнтська та інтерфейс адміністратора.