

**Чернівецький національний університет імені Юрія Федьковича**

(повне найменування закладу вищої освіти)

**факультет математики та інформатики**

(назва інституту/факультету)

**Кафедра математичного моделювання**

(назва кафедри)

## **СИЛАБУС**

**навчальної дисципліни**

**Фреймворки JavaScript**

**вибіркова**

**Освітньо-професійна програма** «Інформаційні технології та управління проектами»  
(назва програми)

**Спеціальність** 122 - Комп'ютерні науки  
(вказати: код, назва)

**Галузь знань** 12 - Інформаційні технології  
(вказати: шифр, назва)

**Рівень вищої освіти** перший (бакалаврський)  
(вказати: перший (бакалаврський)/другий (магістерський)/третій (освітньо-науковий))

факультет математики та інформатики  
(назва факультету/інституту, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

**Мова навчання** українська  
(вказати: на яких мовах читається дисципліна)

**Розробники:** Перцов А.С., доцент, кандидат фізико-математичних наук  
(вказати авторів (викладач (ів)), їхні посади, наукові ступені, вчені звання)

**Профайл викладача (-ів)** <http://matmod.fmi.org.ua/pro-kafedru/spivrobotnyky/pertsov-andriy-sergiyovych/>

**Контактний тел.** (0372)584825

**E-mail:** [a.pertsov@chnu.edu.ua](mailto:a.pertsov@chnu.edu.ua)

**Сторінка курсу в Moodle** <https://moodle.chnu.edu.ua/enrol/index.php?id=3866>

**Консультації** Онлайн-консультації за домовленістю

## 1. Анотація дисципліни

## 2. Мета навчальної дисципліни

**Мета:** вивчення сучасних фреймворків для розробки клієнтської частини Веб-додатків.

Предметом вивчення навчальної дисципліни є вивчення сучасних можливостей розробки клієнтської частини веб-додатків, охоплюючи HTML/CSS, JavaScript/JQuery, та таких фреймворків як AngularJS та Bootstrap.

**Завдання:** Основними завданнями вивчення дисципліни є освоєння студентами основних можливостей роботи з HTML, CSS та JavaScript, набути навички з використання сучасних Web UI фреймворків та інструментів, познайомитися із можливістю розробки клієнтської частини використовуючи JavaScript фреймворку AngularJS.

## 3. Пререквізити.

## 4. Результати навчання:

### знати:

- необхідні елементи для розробки Веб-сторінок, включаючи HTML, CSS and JavaScript,
- клієнтські Веб-фреймворки, зокрема Bootstrap,
- CSS препроцесори, Less та Sass,
- основи Node.js та NPM, а також веб-інструменти, зокрема Bower,
- фронт-ендові фреймворки, які базуються на Javascript, зокрема AngularJS,
- методи розробки односторінкових додатків (SPA) з використанням Angular.

### вміти:

- створювати Веб-сторінки, використовуючи HTML елементи
- застосовувати CSS (style sheet rules) до складових частин веб-сторінок, для зміни поведінки та відображення
- програмувати інтерактивний JavaScript на веб-сторінках,
- налаштовувати, створювати дизайн і стиль веб-сторінки за допомогою Bootstrap та його компонентів
- створювати адаптивний дизайн веб-сторінок
- використовувати веб-інструменти для настройки та управління веб-сайтами,
- бути знайомими з клієнтським JavaScript фреймворком та шаблоном проектування MVC
- створювати односторінкові додатки в AngularJS
- використовувати різні функції Angular, включаючи директиви, фільтри, контролери, область дії та маршрутизацію,
- створювати функціональні фронтенд веб-додатки з використанням AngularJS.

Дисципліна формує такі **компетентності за ОП «Інформаційні технології та управління проєктами»:**

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.

ЗК6. Здатність вчитися й оволодівати сучасними знаннями.

ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК8. Здатність генерувати нові ідеї (креативність).

ЗК9. Здатність працювати в команді.

ЗК10. Здатність бути критичним і самокритичним.

ЗК11. Здатність приймати обґрунтовані рішення.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

ФК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

ФК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з

відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

ФК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.

ФК16. Здатність реалізовувати високопродуктивні обчислення на основі хмарних сервісів і технологій, паралельних і розподілених обчислень при розробці й експлуатації розподілених систем паралельної обробки інформації.

Наведені результати навчання за відповідною дисципліною співвідносяться із такими **програмними результатами навчання:**

**за ОП «Інформаційні технології та управління проєктами»:**

ПРН5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

ПРН9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

ПРН13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.

## 5. Опис навчальної дисципліни

### 5.1. Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	4	7	4	120	15	–	–	30	75	–	залік

### 5.2. Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин										
	денна форма						Заочна форма				
	усього	у тому числі					усього	у тому числі			
л		п	лаб	інд	с.р.	л		п	лаб	інд	с.р.
<b>Змістовий модуль 1. HTML, CSS та JavaScript</b>											
Тема 1. HTML та CSS.	8	1		1		6					
Тема 2. Основи JavaScript.	9	1		2		6					
Тема 3. Поглиблений JavaScript.	10	2		2		6					
Разом за ЗМ 1	27	4		5		18					
<b>Змістовий модуль 2. Front-End WebUI фреймворки та інструменти.</b>											

Тема 1. Веб-системи на стороні клієнта, Bootstrap.	9	1		2		6					
Тема 2. CSS компоненти Bootstrap.	13	2		4		7					
Тема 3. JavaScript компоненти Bootstrap.	13	2		4		7					
Тема 4. Веб-інструменти.	8	1				7					
Разом за ЗМ 2	43	6		10		27					
<b>Змістовий модуль 3. Front-End JavaScript фреймворки.</b>											
Тема 1. Фронт-енд фреймворки на основі Javascript. AngularJS.	10	1		3		6					
Тема 2. Різні аспекти AngularJS.	16	2		6		8					
Тема 3. Розробка односторінкових програм (SPA)	13	1		4		8					
Тема 4. Клієнт-серверна взаємодія та тестування	11	1		2		8					
Разом за ЗМ 3	50	5		15		30					
<b>Усього годин</b>	<b>120</b>	<b>15</b>		<b>30</b>		<b>75</b>					

### 5.3. Тематика лабораторних занять

№	Назва теми
1	Створення простого сайту засобами HTML та CSS.
2	Javascript.
3	Bootstrap.
4	AngularJS.

### 5.4. Самостійна робота студента (ІНДЗ)

№	Назва теми
1	Повторення та поглиблення знань та вміння використання структур HTML. Застосування каскадних таблиць CSS.
2	Основи програмування на JavaScript. Використання JS у сторінках HTML.
3	Bootstrap, як безкоштовний набір інструментів з відкритим кодом.
4	Завантаження та встановлення. Використання Bootstrap у сторінках розмітки гіпертексту.
5	Angular JS, як фреймворк з відкритим програмним кодом, його застосування. Переваги використання Angular JS при розробці веб додатків.
6	Створення клієнтського Angular JS.

### 6. Система контролю та оцінювання

Засобами оцінювання та демонстрування результатів навчання є:

- лабораторні роботи;

- індивідуальні проекти.

### Форми поточного та підсумкового контролю

Поточний контроль знань відбувається під час лабораторних занять шляхом захисту виконаних завдань кожної лабораторної роботи.

Форма підсумкового контролю: залік.

### Розподіл балів, які отримують студенти для заліку

Поточне тестування та самостійна робота											Залік	Сума
Змістовий модуль №1			Змістовий модуль №2				Змістовий модуль №3					
T1	T2	T3	T1	T2	T3	T4	T1	T2	T3	T4	30	100
5	5	5	5	5	5	5	5	5	10	10		

### Критерії оцінювання результатів навчання з навчальної дисципліни

Лабораторні роботи виконуються студентами індивідуально. Структура лабораторних робіт для усіх студентів однакова.

Кожна лабораторна робота оцінюється з точки зору вчасності та якості виконання. Викладач слідкує за виконанням та оцінює студента під час захисту лабораторної роботи. Мінімальна позитивна оцінка формується на основі оцінок виконаних робіт протягом семестру та результату заліку.

### Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	Оцінка за шкалою ЄКТС	
	Оцінка (бали)	Пояснення за розширеною шкалою
Зараховано	A (90-100)	відмінно
	B (80-89)	дуже добре
	C (70-79)	добре
	D (60-69)	задовільно
	E (50-59)	достатньо
Не зараховано	FX (35-49)	(незадовільно) з можливістю повторного складання
	F (1-34)	(незадовільно) з обов'язковим повторним курсом

### 7. Рекомендована література

1. W3School HTML Element Reference. <http://www.w3schools.com/tags/default.asp>
2. W3School CSS Reference <http://www.w3schools.com/cssref/default.asp>
3. W3School JavaScript Tutorial <http://www.w3schools.com/js/default.asp>.
4. W3School JavaScript Style Guide and Coding Conventions  
[http://www.w3schools.com/js/js\\_conventions.asp](http://www.w3schools.com/js/js_conventions.asp).
5. JavaScript exercises <http://www.ugrad.cs.ubc.ca/~cs101/2013W2/practice-questions/prejavascriptartlab/>
6. JavaScript exercises [http://www.billpegam.com/Javascript/javascript\\_exercises.html](http://www.billpegam.com/Javascript/javascript_exercises.html)
7. JavaScript exercises <http://www.jsexercis.com/>

8. W3School JavaScript Reference <http://www.w3schools.com/jsref/default.asp>
9. Bootstrap Home Page <http://getbootstrap.com/>
10. Bootstrap typography <http://getbootstrap.com/css/#type>
11. The 5 Most Popular Frontend Frameworks Compared <http://www.sitepoint.com/5-most-popular-frontend-frameworks-compared/>
12. Front-end Frameworks v2.5 <http://usablica.github.io/front-end-frameworks/>
13. Bootstrap Grid System <http://getbootstrap.com/css/#grid>
14. The Subtle Magic Behind Why the Bootstrap 3 Grid Works <http://www.helloerik.com/the-subtle-magic-behind-why-the-bootstrap-3-grid-works>
15. What The Heck Is Responsive Web Design?  
<http://johnpolacek.github.io/scrolldeck.js/decks/responsive/>
16. Beginner's Guide to Responsive Web Design <http://blog.teamtreehouse.com/beginners-guide-to-responsive-web-design>
17. Guide to Responsive Web Design <http://blog.teamtreehouse.com/modern-field-guide-responsive-web-design>
18. Accessible Rich Internet Applications (ARIA) <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA>
19. Information Architecture 101: Techniques and Best Practices  
<http://sixrevisions.com/usabilityaccessibility/information-architecture-101-techniques-and-best-practices/>
20. Web Site Information Architecture models <http://webdesignfromscratch.com/website-architecture/ia-models/>
21. What is information architecture? [http://www.steptwo.com.au/papers/kmc\\_whatisinforch/](http://www.steptwo.com.au/papers/kmc_whatisinforch/)
22. Information Architecture Tutorial  
[http://www.webmonkey.com/2010/02/Information\\_Architecture\\_Tutorial/](http://www.webmonkey.com/2010/02/Information_Architecture_Tutorial/)
23. The Difference Between Anchors, Inputs and Buttons <http://davidwalsh.name/html5-buttons>
24. When To Use The Button Element <https://css-tricks.com/use-button-element/>
25. JQuery <http://jquery.com/>
26. W3Schools JQuery <http://www.w3schools.com/jquery/default.asp>
27. Nodejs.org <https://nodejs.org/>
28. Npmjs.com <https://www.npmjs.com/>
29. Node API Documentation <https://nodejs.org/api/>
30. NPM Documentation <https://docs.npmjs.com/>
31. Less Getting Started <http://lesscss.org/>
32. Sass Basics <http://sass-lang.com/guide>
33. Getting Started with Less Tutorial <https://scotch.io/tutorials/getting-started-with-less>
34. Getting Started with Sass Tutorial <https://scotch.io/tutorials/getting-started-with-sass>
35. Bower Website <http://bower.io/>
36. Bower 101 <https://medium.com/@ZaidHanania/bower-101-c0b57322df8>
37. Node, Grunt, Bower and Yeoman - A Modern web dev's Toolkit  
<http://juristr.com/blog/2014/08/node-grunt-yeoman-bower/>
38. Angular Official Website <https://angularjs.org/>
39. Angular Documentation <https://docs.angularjs.org/guide>
40. Imperative vs Declarative Programming <https://netguru.co/blog/imperative-vs-declarative>
41. Design Patterns: Elements of Reusable Object-Oriented Software  
<http://c2.com/cgi/wiki?DesignPatternsBook>

42. Grunt <http://gruntjs.com/>
43. Gulp <http://gulpjs.com/>
44. Node, Grunt, Bower and Yeoman - A Modern web dev's Toolkit  
<http://juristr.com/blog/2014/08/node-grunt-yeoman-bower/>
45. AngularJS Scopes: An Introduction <http://blog.carbonfive.com/2014/02/11/angularjs-scopes-an-introduction/>
46. AngularJS Forms <http://tutorials.jenkov.com/angularjs/forms.html>
47. AngularJS Form Validation <https://scotch.io/tutorials/angularjs-form-validation>
48. AngularJS Form Validation with ngMessages <https://scotch.io/tutorials/angularjs-form-validation-with-ngmessages>
49. Bootstrap Form Validation Done Right in AngularJS <http://blog.yodersolutions.com/bootstrap-form-validation-done-right-in-angularjs/>
50. AngularJS Service / Factory Tutorial with Example <http://viralpatel.net/blogs/angularjs-service-factory-tutorial/>
51. AngularJS–Part 6, Templates <https://lostechies.com/gabrielschenker/2013/12/28/angularjspart-6-templates/>
52. AngularJS Dynamic Templates <http://onehungrymind.com/angularjs-dynamic-templates/>
53. Single Page Apps in depth <http://singlepageappbook.com/>
54. SPA and the Single Page Myth <http://www.johnpapa.net/pageinspa/>
55. Single Page Apps with AngularJS Routing and Templating <https://scotch.io/tutorials/single-page-apps-with-angularjs-routing-and-templating>
56. Pretty URLs in AngularJS: Removing the # <https://scotch.io/quick-tips/pretty-urls-in-angularjs-removing-the-hashtag>
57. json-server <https://github.com/typicode/json-server>
58. Creating Demo APIs with json-server <https://egghead.io/lessons/nodejs-creating-demo-apis-with-json-server>
59. JSON <http://www.json.org/>
60. Making API Calls in AngularJS using Angular's \$http service <http://www.sitepoint.com/api-calls-angularjs-http-service/>
61. Creating a CRUD App in Minutes with Angular's \$resource <http://www.sitepoint.com/creating-crud-app-minutes-angulars-resource/>
62. Consuming RESTful APIs <http://fdietz.github.io/recipes-with-angular-js/consuming-external-services/consuming-restful-apis.html>
63. How to Improve Your REST Calls in Angular With ngResource <http://devdactic.com/improving-rest-with-ngresource/>
64. Jasmine 2.3 <http://jasmine.github.io/2.3/introduction.html>
65. Karma <http://karma-runner.github.io/0.13/index.html>
66. Protractor <https://angular.github.io/protractor/#/>
67. Selenium <http://www.seleniumhq.org/>