

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 «Системний аналіз»

(назва програми)

Спеціальність 124 - Системний аналіз

(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)

<https://mathmod.chnu.edu.ua/pro-nas/spivrobotnyky/pertsov-andrii-serhiiovych/>

Контактний тел. (0372)584825

E-mail: 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:

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

вміти:

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

Дисципліна формує такі **компетентності**

ЗК02. Здатність застосовувати знання у практичних ситуаціях

ЗК04. Знання та розуміння предметної області та розуміння професійної діяльності

ЗК05. Здатність спілкуватися державною мовою усно і письмово

ЗК11. Здатність генерувати нові ідеї (креативність)

ФК6. Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних.

ФК7. Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань.

ФК8. Здатність організувати роботу з аналізу та проектування складних систем, створення відповідних інформаційних технологій та програмного забезпечення.

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

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.
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
7. JavaScript exercises <http://www.jsexercises.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_whatisinfoarch/
22. 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/>