COURSE DESCRIPTION

1. Information about the programme

1.1 Institution of higher education	**Alexandru Ioan Cuza University of Iasi**
1.2 Faculty	**Faculty of Economics and Business Administration**
1.3 Department	**Department of Accounting, Information Systems and Statistics**
1.4 Field of study	**Business Informatics**
1.5 Level	**Master**
1.6 Study programme/ Qualification	**Software Development and Business Information Systems**

2. Information about the course

2.1 Course name	**Web Systems Development**						
2.2 Course coordinator	**Associate Prof. Sireteanu Napoleon-Alexandru, Ph.D.**						
2.3 Seminar coordinator	**Associate Prof. Sireteanu Napoleon-Alexandru, Ph.D.**						
2.4 Year of study	II	2.5 Semester	I	2.6 Type of assessment	P	2.7 Discipline status	**C**

*C – Compulsory / * E - Elective*

3. Total estimated time (hours alloted to didactic activity per semester)

3.1 Total number of hours per week	3	of which: 3.2 lecture	2	3.3 seminar/lab	1
3.4 Total number of hours in the curriculum	42	of which: 3.5 lecture	28	3.6 seminar/lab	14

Time distribution	hours
Study of the handbook, coursebook, bibliography and notes	30
Additional research in the library, online and on the field	15
Preparation of seminars/labs, homeworks and projects	40
Tutorials	15
Assessment	8
Other activities..................................	

3.7 Total number of self-study hours	108
3.9 Total number of hours per semester	150
3. 10 Number of credits	6

4. Prerequisites (if applicable)

4.1 curriculum-based	• Programming Languages (or similar), Computer Network (or similar)
4.2 competence-based	• Not applicable

5. Conditions (if applicable)

5.1. for lectures	• Lecture rooms shall be provided with video projector

	• Students will attend lectures. Cell phones must be turned off.
5.2. for seminars/labs	• IT services of the faculty will provide Node.js and Visual Studio Code • Students are invited to bring and use their own laptops: Node.js and Visual Studio Code Labs will have enough computers for students not owning a laptop

6. Assimilated specific competences

Professional competences	• C3.1 Detailed understanding of modern, multi-tiered and service oriented information architectures in order to develop and implement business applications (1) • C3.3 Choose and adapt different commercial and open-source solutions in order to fulfill organizational requirements and which are suited to the organizational constraints (3) • C4.1 Gaining detailed knowledge on all aspects of methodological and technological regarding the representation and persistence of data formats, the protocols and means of communication and integration of applications and services within distributed business information systems (1.5)
Transversal competences	• CT3 – Continuous improvement of specific skills and knowledge towards approaching information systems, development of new software technologies and management of information systems. (0.5)

7. Discipline objectives (provided by the assimilated specific competences grid)

7.1 The general objective of the discipline	Acquiring knowledge and skill in areas of development of React applications
7.2 Specific objectives	• **Component-Based Development**: React is all about building reusable components. Competent React developers should be able to create and manage components effectively, understanding the component lifecycle and state management. • **JSX**: Competent developers should be comfortable with JSX, which is a syntax extension for JavaScript that allows you to write HTML elements and components in a more readable format. • **State Management**: Understanding state management in React is crucial. Competent developers should be familiar with React's built-in state management and might also be skilled in using external state management libraries like Redux or Mobx. • **Routing**: Competent React developers should know how to set up client-side routing in React applications using libraries like React Router. • **Hooks**: React Hooks are a powerful feature for managing state and side effects in functional components. Proficiency with hooks like useState, useEffect, useContext, etc., is important. • **API Integration**: React is often used to build dynamic web applications, so competent developers should be able to make API requests and handle data asynchronously, typically using technologies like Axios or the Fetch API. • **Component Styling**: Understanding how to style React components is important. Competent developers may use CSS, SASS, or CSS-in-JS libraries like styled-components or Emotion.

	• **Testing**: Proficiency in testing React components is valuable. This includes knowledge of testing libraries like Jest and testing utilities like React Testing Library. • **State Management Libraries**: Familiarity with popular state management libraries like Redux or Mobx can be a valuable competency, especially for larger applications. • **Performance Optimization**: Competent React developers should be able to optimize the performance of their applications, understanding concepts like memoization, lazy loading, and minimizing re-renders. • **Debugging**: Proficiency in debugging React applications is crucial. This includes using browser developer tools, React DevTools, and other debugging techniques. • **Version Control**: Being able to work with version control systems like Git for collaborative development is an essential skill. • **Build and Deployment**: Knowing how to build and deploy React applications, whether using Create React App, Webpack, or other tools, is a key competency. • **Error Handling**: Handling errors gracefully in a React application is important. Competent developers should know how to implement error boundaries and handle exceptions. • **Security**: Understanding and implementing security best practices, such as protecting against cross-site scripting (XSS) attacks, is crucial for React development.

8. Content

8. 1 Lecture	Teaching methods	Observations
Chapter 1. Graphical interfaces	PPT presentation, explanation, conversation, questioning.	1 lecture
Chapter 2. JavaScript	PPT presentation, explanation, conversation, questioning.	1 lecture
Chapter 3. React	PPT presentation, explanation, conversation, questioning.	1 lecture
Chapter 4. React. Creating the necessary components	PPT presentation, explanation, conversation, questioning.	1 lecture
Chapter 5. React. JSX Coding Rules of React Components	PPT presentation, explanation, conversation, questioning.	1 lecture
Chapter 6. React-Bootstrap	PPT presentation, explanation, conversation,	1 lecture

	questioning.	
Chapter 7. React. *props* Object	PPT presentation, explanation, conversation, questioning.	1 lectures
Chapter 8. React. State object	PPT presentation, explanation, conversation, questioning.	2 lectures
Chapter 9. React Router	PPT presentation, explanation, conversation, questioning.	1 lecture
Chapter 10. React Component Libraries (Material UI)	PPT presentation, explanation, conversation, questioning.	2 lectures
Chapter 11. React applications in serverless architecture	PPT presentation, explanation, conversation, questioning.	1 lecture
8. 2 Seminar/lab	Teaching methods	Observations
Creating your first React app	Practical Case Discussion	1 lab
Creating the <Book /> Component	Practical Case Discussion	1 lab
Using the <Card /> component to describe the <Book />	Practical Case Discussion	1 lab
<BookList /> Component	Practical Case Discussion	1 lab
Forms. <Add /> Component	Practical Case Discussion	1 lab
Sending to <App /> the state object from <Add />	Practical Case Discussion	1 lab
React Developer Tools	Real-world Exam Case	1 lab
References: Anthony Accomazzo, **Fullstack React: The Complete Guide to ReactJS and Friends**, 2017 **HTML To React:** The Ultimate Guide, NG, 2018 Robin Wieruch, **The Road to React: The React.js with Hooks in JavaScript Book (2023 Edition),** 2023 Mark Thomas, **React in Action,** Manning, 2018		

9. Corroboration of the discipline content with the expectations of epistemic community representatives, professional associations as well as of representative employers in the programme related field.

- The content of this discipline has been decided upon by taking into account both the curricula of some prestigious Western Universities and the demands of the economic environment provided by potential employers, either in the public or in the private IT companies.

10. Assessment

Type of activity	10.1 Assessment criteria	10.2 Assessment methods	10.3 Share of final grade
Grid Test Evaluation			20%
Pratical project	React Project	Presentation, discussion	80%
10.6 Minimum performance standard			
• Setting up three virtual machines and communications between them			

Date of completion
27.09.2023

Lecture Coordinator
Assoc.Prof. Napoleon-Alexandru Sireteanu, Ph.D.

Seminar Coordinators
Assoc.Prof. Napoleon-Alexandru Sireteanu, Ph.D.

Date of approval within the department

Head of Department
prof.univ.dr. Mircea ASANDULUI