**Course Syllabus**

|  |
| --- |
| **CNU International Summer Session** |

|  |  |
| --- | --- |
| **Course Title**  | **Web Programming and Practice** |
| **Course Type** |  | **Credits****(hours)** | 45 |
| **Department** | Computer Science | **Professor** | Assoc. Prof. Dr. CRISAN Daniela Alexandra (Ms) |
| **Classification****(year in school)** | Undergraduate II | **Course Code** | STT3015 |
| **Class room** | TBA | **E-mail** | daniela.alexandra.crisan@rau.ro |
| **Prerequisite(s)**  | Programming languages |
|  |  |  |  |  |  |
| **Course objectives** | Gain comprehensive knowledge of .NET Core Blazor framework for modern web development.  |
| **Course Summary** | This course provides an in-depth exploration of building modern, database-connected websites using Microsoft technologies. Participants will engage with the .NET Core Blazor framework and learn to utilize C#, LINQ, and Microsoft SQL Server to create dynamic, interactive web applications. The course emphasizes productivity-boosting tools such as Entity Framework (ORM) and scaffolding, allowing for more efficient development processes. With a focus on both foundational and advanced concepts, students will work on a project-based approach, culminating in the creation of a live, fully functional website. By the end of the course, participants will possess the technical skills to design, develop, and deploy high-quality web applications. |
| **Teaching Methods** | **Teaching Methods** |
| Lecture | Presentation/Discussion | Problem Based Learning | Project Based Learning | Flipped Learning | Experiment/ Practices | Others(Describe) |
| X | X | X | X | X | X |  |
|  |
| **Grading** | Mid-Term | Final project | Individual Tasks | Team Projects | Class participation | Attendance | Others(Describe) | **Total** |
| **20%** | **40%** | **20%** | **-** | **20%** | **-** | **-** | **100%** |
| ※ Pursuant Section 28 of the Guidelines on Class Management, grading methods can be adjusted for the physically impaired. ※ Under Section 29 of the University Regulations on Academic Affairs, a student automatically fails a course in case of failure to attend more than 3/4 classes. (More than four(4) times absence) |
| **Accommodations for Handicapped**  | - Visually impaired: provision of course related materials in audio, note taking helper, permission to record the lecture- Audibly impaired: provision of course related materials in visual, note taking helper, permission to have e-learning lectures in sign language or shorthand- Physically or mentally challenged: provision of course related materials, note taking helper, permission to record the lecture* Any other requests that are considered necessary: provision of assisted

 ingress and egress to the classrooms and other supports |
| **Textbooks & References** |
| Category | Title | Author | Publisher | Year of publication |
| Main textbook | Web Development with Blazor: A practical guide to building interactive UIs with C# 12 and .NET 8 3rd ed. Edition |  Jimmy Engström | Packt Publishing | 2024 |
| Others | C# 13 and .NET 9 – Modern Cross-Platform Development Fundamentals: Start building websites and services with ASP.NET Core 9, Blazor, and EF Core 9 |  Mark J. Price | Packt Publishing | 2024 |
| Reference | **Online Documentation & Tutorials:** * **Microsoft Learn: Blazor** – Official documentation for learning Blazor, with tutorials on building applications.

[ https://learn.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-7.0]* **Microsoft Docs: Entity Framework Core** – A comprehensive guide on using EF Core for ORM and database operations.

[https://learn.microsoft.com/en-us/ef/core/]* **Microsoft Learn: ASP.NET Core** – Core learning resources for ASP.NET Core, relevant to integrating web applications with .NET.

[https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0]* **YouTube: Blazor Crash Course** – Free tutorial series for beginners to get started with Blazor development.

[https://www.youtube.com/watch?v=Vh3eIHZ2D5A]**Blogs and Articles*** **Dev.to: Blazor Articles** – A collection of blog posts and tutorials on Blazor from developers in the community.
* **Medium: Blazor Development** – Insights, tips, and advanced concepts written by industry experts.

**Tools & Resources*** **Visual Studio Community**– The primary IDE for .NET development, including Blazor projects.

 [https://visualstudio.microsoft.com/vs/community/]* **SQL Server Management Studio (SSMS)** – A robust tool for managing and developing SQL Server databases.

 [https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16] |

|  |
| --- |
| **Daily Course Schedule** |
|  |

| **Day** | **Lecture Topic** | **Hours per day** | **Method of Instruction** | **Class Materials & Assignments** |
| --- | --- | --- | --- | --- |
| 1. | **Introduction to .NET Core and Blazor Framework*** Overview of .NET Core and its advantages for web development.
* Introduction to Blazor and its architecture.
* Setting up the development environment (Visual Studio).
 | 3 | Lecture, Presentation/Discussion | Slides, Visual Studio setup guide, architecture diagrams. |
| 2. | **Building a Basic Blazor Web Application*** Creating your first Blazor WebAssembly app.
* Understanding the project structure and main components.
* Using components and managing application routes.
 | 3 | Experiment/Practice, Project-Based Learning | Step-by-step tutorial, project templates, Visual Studio examples. |
| 3. | **Fundamentals of C#*** Review of C# language basics relevant for web development.
* Practical exercises with C# in the context of web applications.
* Writing simple Blazor components.
 | 3 | Lecture, Problem-Based Learning | C# problem sets, code snippets, quick syntax guide. |
| 4. | **Advanced Blazor Components and Data Binding*** Creating reusable components and understanding data binding.
* Implementing event handling and forms in Blazor.
* Data validation with Blazor.
 | 3 | Experiment/Practice, Problem-Based Learning | Component examples, sample projects, Blazor validation docs. |
| 5. | **Relational databases and Microsoft SQL Server*** Overview of Microsoft SQL Server as a relational database management system.
* Designing a database schema for a web application.
* Importance of relational data modeling in SQL Server.
 | 3 | Lecture, Presentation/Discussion | Diagrams, schema designs, SSMS setup guide. |
| 6. | **Integrating Microsoft SQL Server with .NET Core*** Setting up and connecting to a SQL Server database.
* Creating database models and working with Entity Framework Core (EF Core).
* Implementing data access logic in Blazor applications.
 | 3 | Experiment/Practice, Lecture | EF Core examples, database models, Entity Framework guides. |
| 7. | **Using Scaffolding to Accelerate Development*** Understanding the scaffolding process and its benefits.
* Generating code with scaffolding tools.
* Customizing scaffolded components for specific requirements.
 | 3 | Experiment/Practice, Project-Based Learning | Scaffolded code examples, tutorials, customization guides. |
| 8. | **Written test** | 1 | Written test |  |
| **Object-Relational Mapping (ORM) with EF Core*** Deep dive into EF Core and its core features.
* Understanding EF Core migrations.

Best practices for data handling and performance optimization. | 2 | Lecture, Experiment/Practice | EF Core migration exercises, performance optimization checklists. |
| 9. | **Fundamentals of LINQ*** Introduction to LINQ and its use for data manipulation.
* Writing queries with LINQ for specific requirements.
 | 3 | Lecture, Problem-Based Learning | LINQ query examples, interactive simulators, reference materials. |
| 10. | **Review on CRUD scaffolded components*** Reviewing how Create, Read, Update, and Delete functionalities are scaffolded.
* Customizing Scaffolded Code using LINQ queries to meet specific application requirements.
 | 3 | Presentation/Discussion, Experiment/Practice | Pre-scaffolded CRUD examples, LINQ exercises, customization scenarios. |
| 11. | **Implementing Authentication and Authorization*** Integrating authentication with ASP.NET Core Identity.
* Managing user roles and permissions.
* Securing routes and components in Blazor applications.
 | 3 | Lecture, Experiment/Practice | ASP.NET Core Identity guides, role-based examples, securing routes exercises. |
| 12. | **Creating a Responsive UI with Blazor*** Implementing responsive design principles for a modern web app.
* Using CSS and Blazor styling tools.
* Enhancing UI/UX with Blazor components and JavaScript interop.
 | 3 | Flipped Learning, Experiment/Practice | Video tutorials, example projects, UI framework documentation (e.g., Bootstrap). |
| 13. | **Deploying Your Blazor Application*** Preparing the app for deployment (build configurations, production settings).
* Deploying Blazor WebAssembly apps to a web server or cloud platform.
* Continuous integration/continuous deployment (CI/CD) overview.
 | 3 | Lecture, Experiment/Practice | Deployment guides (IIS, Azure), CI/CD examples, troubleshooting tips. |
| 14. | **Synthesizing Knowledge and Preparing for the Final Project*** Comprehensive review of major course topics, including:
	+ Building and structuring Blazor applications.
	+ Database integration and data handling with EF Core and LINQ.
	+ Authentication, authorization, and security.
	+ Responsive UI design and deployment best practices.
* Open discussion to clarify lingering doubts and revisit challenging concepts.
* Brainstorming session: Final project scope and integration strategies.
 | 3 | Interactive discussion and Q&A, Group brainstorming and planning exercises, Collaborative review of examples and past assignments. | Summary slides covering key lessons, Case studies demonstrating integrated applications. |
| 15. | **Capstone Project and Course Review*** Final project presentation.
* Discussion on advanced topics for further learning.
* Final review and feedback session.
 | 3 | Presentation/Discussion | Presentations, advanced learning resources. |

|  |
| --- |
| **References** |
| **Daily Course Contents**The course consists of 15 lessons covering topics like Blazor framework basics, C# programming, database integration, scaffolding, ORM, LINQ, authentication, and deployment. Each lesson introduces new concepts and includes hands-on exercises to reinforce learning.**Teaching Methods**A mix of instructional techniques is used, including interactive lectures, hands-on practice, project-based learning, and flipped classroom strategies. Students work on incremental assignments culminating in a capstone project, integrating all learned skills.**Assignments**Students complete small coding exercises after each lesson, build a database schema for their applications, implement CRUD operations using Blazor, and develop a final project, which involves creating a web application. **Student Evaluation Methods**Evaluation is based on participation (20%), coding exercises (20%), a mid-term project (20%), and the final project (40%). The mid-term evaluation consists in a one hour written test (multiple-choice questions) that will assess students' understanding of the concepts taught in the first 7 classes. Sample questions will be provided in the previous lessons to help students prepare. The test is worth 20% of the final grade.The final project is an individual database-driven Blazor application that will integrate all learned concepts. The application will be presented in the class, in order to demonstrate the student’s understanding and implementation of key concepts. The project is worth 40% of the final grade.Note: In addition, with the Chonnam National University's consent, interested participants to the course will receive guidance on publishing their project as a scientific article in the ***Journal of Computer Sciences Applied in Management***, a review for students’ scientific research published yearly by the Romanian-American University from Bucharest, Romania. More information can be found at: *https://web.rau.ro/websites/riam/*.  |