

## Course Specification Document

|              |                              |
|--------------|------------------------------|
| <b>Title</b> | Web Applications Development |
|--------------|------------------------------|

|                |        |
|----------------|--------|
| <b>Credits</b> | 5 ECTS |
|----------------|--------|

|             |  |
|-------------|--|
| <b>Aims</b> | This course aims to support the student technically in the field of web applications by providing them with essential and important knowledge in building web applications. This enables him to use this knowledge when needed in other cognitive domains. |
|-------------|--|

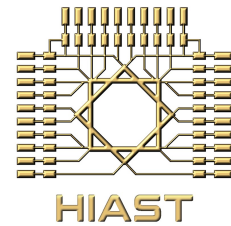
### Intended learning outcomes

On successful completion of this course, the student will be able to:

- Understand the fundamental concepts of the web.
- Comprehend the mechanism of the web, the life cycle of a web page request.
- Familiarize himself with the methods of building basic and advanced web applications and linking them with databases.
- Identify key considerations in designing web applications and know general solutions to common problems.
- Know the principles of designing attractive, interactive web application interfaces suitable for different screens.
- Understand web services.
- Grasp concepts of e-commerce.
- Utilize caching techniques, multilingualism, state management, and identity authentication when building advanced web applications.
- Benefit from communication mechanisms with databases and linking between relational spaces in databases and purpose spaces in object-oriented programming languages.
- Develop agile web applications through optimal use of data, programming languages, and various frameworks within architectural structures, based on objective feature comparisons.

### Syllabus

- **Introduction:** Client-Server system, HTTP protocol, web servers.
- **Client-Side technologies:** HTML, CSS, Javascript – DOM.
- **Forms:** Form inputs, form actions, form methods.
- **State management:** Meaning of state management, Session, Cookie, Tokens, local storage.
- **Database Access - ORM:** Traditional database access patterns and query execution - Native SQL queries, ORM model and identity, Granularity, Subtypes (Inheritance), associations, data navigation, data loading (Lazy vs. Eager).
- **MVC model:** MVC architectural model, Model binding, Pass data to views, View engines.



- **Middleware:** Definition and function of middlewares, how to benefit from existing middleware layers, how to define a new middleware layer.
- **Error handling:** Handling Expected exceptions in the application, handling unexpected errors.
- **Web services:** Introduction to Web services, REST, SOAP.
- **Authentication & Authorization:** Authentication (Session-Based vs Token-Based), authorization models (RBAC, ABAC, CBAC).
- **Internationalization and Localization:** Application-Level Multilingualism, Database-Level multilingualism, handling date differences based on language and culture.
- **Caching:** Meaning of caching, possible places for storing cached copies, managing cached copies.