

| | |
|----------------------------|---|
| Module | Server Side Web Development |
| Course code | BSCH-SWD |
| Credits | 5 |
| Important notes | Please note that this module is not intended for first year students. |
| Allocation of marks | 20% Continuous Assessment |
| | 80% Project |

Intended Module Learning Outcomes

On successful completion of this module learners will be able to:

1. understand how server-side dynamic web pages are delivered to end-users.
2. design and build dynamic database-backed web sites.
3. build a state-based user experience on top of stateless protocols
4. recognize security issues in web development and suggest and implement best practice solutions.

Module Objectives

This module introduces the learner to the fundamentals behind server-side web development. They are introduced to the core concepts behind dynamic, database driven web development, through server-side scripting and database integration and learn how to design and build web applications that deliver database information through server-side HTML preprocessing.

Learners are given practical experience of developing dynamic web sites using these technologies.

Module Curriculum

Web Application Development

- Web architecture
- Client-Server Relationships
- Three-tier applications
- Web Applications
- GET/POST
- Security

Server-side Programming

- Web scripting (PHP)
- processing form data
- validation
- state management (cookies/sessions)
- Security

Integrating Databases

- Database connectivity
- Security

- Propositional and Predicate Calculus;
- rules of inference;
- assertions over sequences;
- trading laws; quantifiers - \forall , \exists , $\exists!$, $+$, $*$, max, min -
- reasoning with quantifiers;
- Set notation;
- axioms and set operators, set theorems;
- theory of bags;
- theory of sequences;
- binary relations;
- relations: domain restriction, range restriction, domain subtraction, range subtraction, relational image, relational inverse, composition of relations;
- functions, partial functions, total functions, injective, surjective, bijective function;
- writing simple specifications with sets, relations, functions and sequences;
- using definitions to prove simple properties of specifications.

The *Perfect Developer* specification notation

- Properties of state based specification notations. Examples of state based specification notations.
- The *Perfect Developer* notation: abstract machines, parameterized machines;
- states, constraints on states;
- data types;
- data structures - sets, relations, functions, arrays;
- statements - skip, if..else, case .., invariants, choice, select, pre, post, sees, uses, includes;
- multiple inclusion.
- Case studies.
- Proof obligations.
- Data refinement.
- Design by Contract.
- Implementation of design by contract in *Perfect Developer*.
- Implementing specifications with design by contract.