Web Application Development (ACS-305)

Course Overview

Module Code	2025-ACS-305
ECTS	5
Semester	5th (BSc Computer Science)
Format	14 weeks, 2 × 75 min sessions per week (28 sessions)
Workload	125 hours total
Coordinator	Prof. Dr. Alexander Omelchenko

What You Will Build

An **AI Chat Application** — a full-stack app that lets you talk to AI models. You will build it from scratch across three platforms:

- **Web app** — Next.js, deployed as an installable PWA on Vercel
- **Android app** — Kotlin with Compose Multiplatform
- **iOS app** — same Kotlin codebase, built via Mac pairing

The app connects to free AI models via OpenRouter, or to locally running models via Ollama.

Technology Stack

Layer	Technology
Web Frontend	React, Tailwind CSS, Zustand, TanStack Query
Web Framework	Next.js (App Router)
Backend	Next.js API Routes (serverless)
Database	Prisma Postgres (via Vercel)
ORM	Prisma
Mobile	Compose Multiplatform (Kotlin)
AI	OpenRouter (cloud) / Ollama (local)
Hosting	Vercel

Tools & Software

- **VS Code** — primary editor for web development (weeks 1–9)
- **Android Studio** — for mobile development (weeks 10–12)
- **Git & GitHub** — version control and assignment submission
- **Node.js** (LTS) — JavaScript runtime

- **Ollama** — local AI model runner (weeks 13–14)

Assessment

Component	Weight	Scope
Written Exam (120 min)	50%	HTML, CSS, JavaScript, React, Next.js, Prisma
Project Assignments (10 × 5%)	50%	Cumulative project deliverables

Exam scope: Local Next.js application development. No mobile, deployment, or infrastructure topics.

Assignments: Each builds on the previous one — you develop one project throughout the semester.

#	Deliverable	Due
A1	Git repo with ESLint + Prettier configured	Week 1
A2	Static chat UI prototype (HTML + Tailwind)	Week 2
A3	Interactive chat with OpenRouter streaming (plain JS)	Week 4
A4	Next.js app with server-side API route	Week 6
A5	Chat persistence with Prisma	Week 7
A6	Deployed on Vercel with database	Week 9
A7	Installable PWA	Week 9
A8	Compose Multiplatform chat UI calling Vercel API	Week 11
A9	Working Android APK	Week 12
A10	Production builds with environment switching	Week 14

Accounts You Will Need

Create these before or during the first weeks:

- **GitHub** account (free)
- **OpenRouter** account (free — for AI API access)
- **Vercel** account (free — linked to GitHub)

Weekly Schedule

Phase 1: Foundations & Prototyping (Weeks 1–2)

Week	Session	Topic
1	1	Course introduction, project overview, Git fundamentals
1	2	VS Code, ESLint, Prettier, EditorConfig, HTML & DOM
2	3	CSS fundamentals, responsive design, Tailwind CSS

2	4	Building the chat UI prototype (static HTML + Tailwind)

Phase 2: JavaScript & Interactivity (Weeks 3–4)

Week	Session	Topic
3	5	JavaScript: ES6+ syntax, async/await, Promises, modules
3	6	Fetch API, streaming responses, calling OpenRouter from the browser
4	7	React: components, JSX, props, state, event handling
4	8	React: useEffect, lifting state, custom hooks, rebuilding chat in React

Phase 3: Next.js & Persistence (Weeks 5–7)

Week	Session	Topic
5	9	Next.js: App Router, routing, layouts, Server vs Client Components
5	10	API routes, server-side OpenRouter proxy, environment variables
6	11	Prisma: schema, migrations, Prisma Client, CRUD in API routes
6	12	Full chat persistence: conversations, messages, history
7	13	State management: Zustand (UI state), TanStack Query (server state)
7	14	Polish: loading states, error handling, optimistic updates

Phase 4: Deployment & PWA (Weeks 8–9)

Week	Session	Topic
8	15	Vercel deployment, Prisma Postgres setup, preview environments
8	16	Environment configuration: local, preview, production
9	17	PWA: manifest, service worker, offline support, installability
9	18	PWA testing on mobile devices, push notifications overview

Phase 5: Mobile with Compose Multiplatform (Weeks 10–12)

Week	Session	Topic
10	19	Kotlin: syntax, null safety, data classes, coroutines. Android Studio setup
10	20	Jetpack Compose: composables, state, layouts, Material 3
11	21	Compose Multiplatform: project structure, shared code, Gradle config
11	22	Chat UI in Compose, Ktor HTTP client, calling Vercel API
12	23	Streaming in Ktor, ViewModel pattern, error handling

12	24	Building: Android APK, iOS via Mac pairing, testing on devices

Phase 6: Local AI & Production (Weeks 13–14)

Week	Session	Topic
13	25	Ollama: local AI models. Cloudflared: tunneling for mobile access
13	26	Environment management across platforms (Next.js, Android, iOS)
14	27	Production: signed APK, TestFlight, Vercel production deployment
14	28	Final presentations and code review

Learning Outcomes

By the end of this course, you will be able to:

1. Explain the document object model behind HTML and its relation to CSS
2. Discuss the principles and basic mechanisms of reactive website design
3. Analyze the interactions between web applications and web services
4. Use Kotlin to implement mobile applications with Compose Multiplatform
5. Use HTML, CSS, and JavaScript to implement web applications running in standard web browsers

Literature

- Stoyan Stefanov: *JavaScript Patterns*, O'Reilly Media, 2010
- Alexey Soshin: *Hands-on Design Patterns with Kotlin*, Packt Publishing, 2018
- Alex Banks, Eve Porcello: *Learning React*, O'Reilly, 2017
- Official documentation: Next.js, React, Prisma, Tailwind CSS, Kotlin