

Veröffentlichung einer Ausschreibung CeDiS 500,- bis 10.000,-
Ausschreibung für ein Werk bis maximal € 10.000,00 (netto):

Implementation eines Theme-Plugins für OJS 3

ID der Ausschreibung: EP-2019-02-QIO

Auftraggeber:

Freie Universität Berlin
Universitätsbibliothek
Center für Digitale Systeme (CeDiS)
Innestr. 24
14195 Berlin

Angebot ist zu richten an:

Freie Universität Berlin
Universitätsbibliothek
Center für Digitale Systeme (CeDiS)
Innestr. 24
14195 Berlin

Mit dem Angebot sind die folgenden Ausschreibungsunterlagen einzureichen:

Eigenerklärung Bieterreignung (gemäß § 6 Abs. 3 VOL/A)

(Die „Zusätzlichen Vertragsbedingungen der Freien Universität Berlin“ sind Vertragsbestandteil: <https://www.fu-berlin.de/zvb>)

Art der Vergabe:

freihändig

Form, in der das Angebot einzureichen ist:

- schriftlich (postalisch)
- per Fax
- elektronisch (per Mail an: cedis@fu-berlin.de)

Art und Umfang der Leistung:

Open Journal Systems (OJS) ist eine Open-Source-Software für die Verwaltung und Veröffentlichung von wissenschaftlichen Zeitschriften. Die Software verfügt u.a. über Theme-Plugins, die den Redaktionen dieser Journals die Möglichkeit zur selbständigen Design-Gestaltung geben.

Im Rahmen des hier ausgeschriebenen und zu erstellenden Auftrages ist ein **OJS-Theme-Plugin** gemäß einer vorliegenden Spezifikation zu implementieren.

Das Theme-Plugin für OJS 3 ist als Child Theme des Default Themes zu implementieren; es soll es OJS-Manager/innen erlauben, über die OJS-Benutzer-Oberfläche im OJS-Backend eigenständig Design-Anpassungen vorzunehmen.

Das Theme-Plugin soll so implementiert werden, dass Wartung und Erweiterung möglichst wenig Aufwand erfordern.

Elemente des Auftrages:

TAP 1: Erstellung eines Grunddesigns und Einbindung des Journal Stylesheet

Das Design soll mittels *LESS* umgesetzt und im Theme-Plugin hinterlegt werden, damit die im Backend gemachten Einstellungen über *LESS-Variablen* automatisch eingefügt werden können. In Zusammenarbeit mit dem Design-Team des Center für Digitale Systeme (CeDiS)/Universitätsbibliothek der Freien Universität Berlin soll eine Frontend-Design-LESS-Datei als Standard-Design erarbeitet und umgesetzt werden, das es Zeitschriftenherausgeber/innen ermöglichen soll, ohne die Erstellung weiterer CSS- oder LESS-Dateien die Funktionen des Theme-Plugins zu nutzen. Das Design soll modernen, responsiven Ansprüchen genügen. Bei der Erstellung des Designs sind die Optionen des Theme-Plugins zu beachten und zu berücksichtigen. Farben, Schriftkonfigurationen und ähnliche Einstellungen müssen über Variablen umgesetzt werden, die über die LESS-Dateien der einzelnen Arbeitspakete gesetzt werden. Für versierte Benutzer/innen besteht die Option, mit einer zusätzlichen LESS-/CSS-Datei tiefer in die Struktur des Designs einzugreifen, um individuelle Anpassungen vornehmen zu können, die nicht über die grafische Oberfläche des Theme-Plugins im Backend unter Appearance möglich sind. Diese LESS-Datei muss zuletzt geladen und referenziert werden, um vorherige CSS-Properties überschreiben zu können.

Zu implementierende Funktionen

- Erstellung eines Grunddesigns
- Einbindung des *Journal Stylesheet*

Technische Umsetzung

- Modularer Aufbau; die Frontend-Design-LESS-Datei referenziert über *@import* die jeweiligen LESS-Dateien der einzelnen Arbeitspakete; hiermit wird die Übersichtlichkeit des Codes gewährleistet und ein späteres Hinzufügen weiterer Funktionen erleichtert, gleichzeitig ist das Überschreiben von LESS/CSS-Properties auf diese Weise einfacher und übersichtlicher zu realisieren.
- Farben und Schriften sind als LESS-Variablen zu implementieren, die über die LESS-Dateien der jeweiligen Arbeitspakete gesetzt werden.
- Das Journal Stylesheet wird im *@import*-Statement als LESS-Datei referenziert, um damit mitkompiliert zu werden; auf diese Weise sind LESS-Funktionen im Journal Stylesheet möglich.

TAP 2: Farben

Die im Rahmen des Default-Theme-Plugins ermöglichte farbliche Anpassung ist nur rudimentär nutzbar, komplexere Farbschemata sind bislang nicht umsetzbar. Das zu erstellende Theme-Plugin soll die farbliche Gestaltung über Key-, Kontrast- und Neutralfarbe ermöglichen. Die hierüber definierten Farben werden mittels LESS-Variablen gesetzt, die in der Frontend-Design-LESS-Datei verwendet werden. Als zusätzliche Option sollen automatisch Farbschattierungen der Keyfarbe erstellt werden können. Zeitschriftenherausgeber/innen sollen über das Backend von OJS die Farbwerte einfach mittels Farbauswahl-Dialog wählen können. Die Zuweisung von Key-, Kontrast- und Neutralfarbe soll in Zusammenarbeit mit dem CeDiS-Design-Team im Zuge der Erstellung des Frontend-Designs erfolgen.

Zu implementierende Funktionen

- Benutzerspezifische Farbschemata auf Basis von Key-, Kontrast- und Neutralfarbe
- Optionale Generierung von prozentualen Schattierungen der Keyfarbe
- Zuweisung von Key-, Kontrast- und Neutralfarbe

Technische Umsetzung

- Bereitstellen einer LESS-Datei zur Farbkonfiguration, die in die Frontend-Design-LESS-Datei geladen wird; sämtliche Farbaufrufe in der Frontend-Design-LESS-Datei erfolgen über LESS-Variablen, die in der genannten Farbkonfiguration gesetzt werden; die prozentualen Schattierungen der Key-Farbe werden bei nicht-gewünschter Schattierung auf den Farbwert der Keyfarbe gesetzt.

- Hinzufügen *einer Checkbox* im Backend unter Appearance zur (De-)Aktivierung der Schattierungsoption.
- Hinzufügen von *drei Textfeldern* zur Definition von Key-, Kontrast- und Neutralfarbe; zur einfacheren Farbauswahl wird den Zeitschriftenherausgeber/innen ein Farbauswahl-Dialog im Backend zur Verfügung gestellt.

TAP 3: Schriften

Das Default-Theme-Plugin bietet eine kleine Auswahl von Schriftarten über das Backend an. Aufgrund datenschutzrechtlicher Bestimmungen ist die *Nutzung der Google Font API nicht gestattet*, sodass eine Auswahl im Rahmen von Stylesheets auf Systemschriften oder auf dem Server verfügbare Schriften begrenzt ist. In Zusammenarbeit mit dem CeDiS-Design-Team soll eine Liste von frei verfügbaren Schriften erstellt werden, die im Rahmen des Plugins bereitzustellen sind. Als Erweiterung der bestehenden Funktionalität soll das zu erstellende Theme-Plugin die Option von Schriftkombinationen bieten. Die Zuweisung erfolgt hier auf Basis von Überschriften und Fließtext. Darüber hinaus soll für die Schriften jeweils eine Standardkonfiguration hinterlegt sein, die Schriftgrade und -schnitte den jeweiligen HTML-Elementen zuweist. Diese Standardkonfiguration soll für Zeitschriftenherausgeber/innen als Empfehlung dienen, die mittels des optionalen User-Stylesheets überschrieben werden kann.

Zu implementierende Funktionen

- Bereitstellung einer Auswahl von Schriften
- Kombination zwei verschiedener Schriften bzw. Schriftschnitte für Fließtexte und Überschriften

Technische Umsetzung

- Schrift-Dateien sind aus datenschutzrechtlichen Gründen in das Plugin zu integrieren.
- Die Schriftkonfiguration erfolgt in einer eigenen LESS-Datei, die in der Frontend-Design-LESS-Datei geladen wird; das Journal Stylesheet wird als letzte Datei geladen, sodass ein Überschreiben mittels einer eigenen CSS-Datei jederzeit möglich ist.
- Die Zuweisung der beiden ausgewählten Schriften zu den zugehörigen HTML-Elementen erfolgt über LESS-Variablen.
- Für jede Schrift ist eine Standard-Schriftgröße als LESS-Variable zu definieren.



- Hinzufügen von *zwei Dropdown-Menüs* im Backend unter Appearance zur Schriftauswahl.
- Hinzufügen *eines Datei-Upload-Dialogs*, um eine LESS-Datei hochladen zu können, die die Schriftkonfiguration im Hinblick auf Schriftgröße, -farbe, -art und -zuweisung überschreibt; diese Datei muss direkt nach der generierten Standard-Konfiguration als *@import* in der Frontend-Design-LESSDatei geladen werden.

TAP 4: Frontend-Header

Das bisherige Frontend-Header-Template des Default-Themes sieht vor, dass Zeitschriftenherausgeber/innen eine Entscheidung zwischen einem Logo- bzw. Headerbild oder einer textlichen Darstellung ihres Zeitschriftennamens treffen müssen. Diese Einschränkung stellt ein Problem für das Erstellen von responsiven Designs dar. Headerbilder skalieren in der mobilen Ansicht zu unleserlichen Dimensionen, ein allgemein übliches Ersetzen eines Headerbildes oder Logos in der mobilen Ansicht durch den Zeitschriftennamen als Text ist nicht in den Templates vorgesehen, ein Umschalten über LESS/CSS ist daher bislang nicht möglich. Im Hinblick auf die Gestaltung von bestehenden OJS2-Zeitschriften der Freien Universität Berlin soll eine Lösung erarbeitet werden, um eine Weiterentwicklung bestehender Designs zu ermöglichen und nicht mit diesen vollständig zu brechen. Das Design-Plugin soll daher standardmäßig sowohl den Zeitschriftennamen als Textelement als auch ein Logo, sofern ein solches hochgeladen wurde, in der HTMLDatei integrieren. Ein Ein- und Ausblenden soll mittels LESS/CSS gelöst werden. Zusätzlich soll die Option bestehen, ein Hintergrundbild für den Headerbereich hochzuladen, um bestehende Gestaltungen von OJS2-Installationen aufgreifen zu können. Das responsive Verhalten entlang der noch zu bestimmenden Umbruchpunkte ist in Zusammenarbeit mit dem CeDiS-Design-Team im Rahmen der Erstellung des Frontend-Designs zu bestimmen.

Zu implementierende Funktionen

- Parallele Bereitstellung von Zeitschriftename und -logo im Header
- Aus-/Einblendung von Zeitschriftename und -logo entsprechend noch zu definierender Breakpoints
- Hinzufügen eines optionalen Header-Hintergrundbildes

Technische Umsetzung

- Erstellung eines eigenen Header-Templates, das über das Theme-Plugin anstelle der bestehenden *header.tpl* in *lib/pkp/templates/frontend/components/* geladen wird; *keine Veränderung des bestehenden Templates.*
- Steuerung der Sichtbarkeit von Zeitschriftename und -logo über das *CSS-Property display.*

- Darstellung des optionalen Header-Hintergrundbildes über das *CSS-Property background*.
- Hinzufügen *eines zusätzlichen Upload-Dialogs* für das Hochladen des Header-Hintergrundbildes im Backend unter Appearance.
- Das optionale Header-Hintergrundbild ist in der OJS-Ordnerstruktur analog dem Zeitschriftenlogo abzulegen.

TAP 5: Journal Startseite

Ähnlich des Bootstrap3 OJS Theme Plugins (<https://github.com/NateWr/bootstrap3/>) soll das zu erstellende Theme Plugin die in OJS vorgesehene „Journal Description“ auf der Startseite eines Journals anzeigen. Bislang wird dieser Text nur auf der Journal-Übersichtsseite angezeigt, die nur dann verfügbar ist, wenn mehr als ein Journal in einer OJS3-Installation eingerichtet ist. Der „Additional Home Content“ stellt damit bislang den einzigen beschreibenden Text dar, allerdings wird dieser erst nach „Homepage Image“, „Announcements“ und „Latest Issue“ angezeigt und kann daher nicht als erste Einleitung dienen. Für die Realisierung muss ein Smarty-Template auf Basis des Standard-Templates „templates/frontend/pages/indexJournal.tpl“ erstellt werden. Die Inhalte werden über bestehende OJS3-Variablen eingebunden.

Zu implementierende Funktionen

- Anzeigen der Journal Description auf der Journal Startseite
- Möglichkeit, den Additional Content-Block vor den Announcements-Bereich zu verschieben

Technische Umsetzung

- Hinzufügen der in OJS3 definierten PHP-Variable „`$journalDescription`“ eingebettet in eine if-Abfrage zur Überprüfung des Vorhandenseins eines Inhaltes.
- Verschieben des bereits bestehenden Implementationsblockes, die die Variable „`$additionalContent`“ in einer `<section>` unterhalb der „`current_issue`“-`<section>` anzeigt; Ziel ist die Darstellung vor der „`cmp_announcements`“-`<section>`.

TAP 6: Borders

Zur einfacheren Gestaltung soll das zu entwickelnde Theme-Plugin eine Option im Backend bereitstellen, die mittels eines DropDown-Menüs die Borders der angezeigten HTML-Objekte entweder ausblendet, teil-ausblendet oder vollständig einblendet. Die visuelle Gestaltung der Borders und die Selektion für das Teilausblenden erfolgt in enger Absprache mit dem CeDiS-Design-Team.

Zu implementierende Funktionen

- Steuerung der Darstellung von Borders der anzuzeigenden HTML-Objekte in den Zuständen „Einblenden“, „Teilausblenden“ und „Ausblenden“
- Steuerung über DropDown-Menü im Backend

Technische Umsetzung

- Hinzufügen eines DropDown-Menüs im Backend mit den Auswahlmöglichkeiten „Einblenden“, „Teilausblenden“ und „Ausblenden“.
- Steuerung der Border-Eigenschaften über das CSS-Property „border-style“.
- Das Teilausblenden wird über LESS-Variablen gesteuert: Es werden zwei Variablen erstellt, je eine für die Borders, die auch beim Teilausblenden angezeigt werden und eine für die, die dabei ausgeblendet werden.
- In Abhängigkeit von der Auswahl im DropDown-Menü wird entweder eine der Variablen (Teilausblenden) oder beide Variablen (Ausblenden) werden auf „none“ gesetzt, oder beide Variablen auf „solid“.

TAP 7: Sidebar

Das Standard-Design der Sidebar im Default-Theme bietet im Gegensatz zu OJS2 nicht mehr die Option, diese an eine der beiden Bildschirmseiten zu verschieben. Das zu erstellende Theme-Plugin soll im Backend ermöglichen, die Sidebar entweder standardmäßig an der rechten Seite darzustellen oder diese zur linken Seite zu verschieben. Mit OJS3 wurde darüber hinaus die Grundlage für ein modernes, responsives Design gelegt, das die Benutzbarkeit auf Mobilgeräten gewährleisten soll. Da das Default-Theme bislang nicht erlaubt, die Sidebar in der Mobilansicht auszublenden, soll im zu erstellenden Theme-Plugin die Möglichkeit geschaffen werden, die Sidebar in der Mobilansicht unter den Footer zu verschieben.

Zu implementierende Funktionen

- Umschaltfunktion im Backend bereitstellen, die das Verschieben der Sidebar ermöglicht
- Verschieben der Sidebar in der Mobilansicht

Technische Umsetzung

- Hinzufügen eines Drop-Down-Menüs zur Auswahl der Darstellungsseite der Sidebar.
- Verschieben der Sidebar mittels CSS über das float-Property.

TAP 8: Hero-Claim

In Anlehnung an einige Seiten der Freien Universität Berlin, die nicht an das Corporate Design gebunden sind, soll das Theme-Plugin die Option eines sog. Hero-Bildes mit zusätzlichem Claim-Text bieten (siehe dazu als Beispiel: <https://www.zibi-berlin.de/>). Über das Backend soll eine Bilddatei hochgeladen werden können, die unterhalb des Header-Bereiches mit dem Logo bzw. dem Zeitschriftennamen und oberhalb der Hauptnavigation über die gesamte Bildschirmbreite angezeigt wird. Über dieses Bild soll der ebenfalls über das Backend eingebbare Text mittig zentriert geschrieben werden (siehe <https://www.zibi-berlin.de/>).

Da der Standard-Header keine Option für Hero und Claim bietet, muss zur Realisierung ein zusätzliches Template erstellt werden, das im Frontend-Header-Template verlinkt werden muss. Die Bereitstellung hat über das Theme-Plugin zu erfolgen. Es ist möglich, den Dateiapload-Dialog für ein optionales Header-Hintergrundbild für den Hero zu benutzen. In diesem Fall ist Zeitschriftenherausgeber/innen die Wahl zwischen keinem Zusatzbild, einem Header-Hintergrundbild oder einem Hero-Bild zu geben.

Das responsive Verhalten entlang der noch zu bestimmenden Umbruchpunkte ist in Zusammenarbeit mit dem CeDiS-Design-Team im Rahmen der Erstellung des Frontend-Designs zu bestimmen.

Zu implementierende Funktionen

- Optionales Hero-Bild über die gesamte Bildschirmbreite zwischen Navigationsleiste und Zeitschriftename/-logo
- Optionaler Claim-Text visuell über dem Hero-Bild
- Nutzung des optionalen Header-Hintergrundbildes als Hero-Bild

Technische Umsetzung

- Erstellung eines *zusätzlichen Smarty-Templates für Hero + Claim*, das mittels `{include ...}` zwischen `<div class="pkp_site_name_wrapper">` und `<nav class="pkp_navigation_primary_row">` in der Theme-Plugin-spezifischen *Header.tpl* geladen wird.
- Ein-/Ausblendung über *CSS-Property display*.
- Anzeige des Hero-Bildes als *CSS-Property background*; Als Datei wird die als optionales Header-Hintergrundbild hochgeladene Datei genutzt, das *CSS-Property background* von `<div class="pkp_site_name_wrapper">` wird gleichzeitig auf *none* gesetzt.
- Hinzufügen *einer HTML-Checkbox* im Backend unter Appearance.

Allgemeine Rahmenbedingungen

- CeDiS benennt eine/n Ansprechpartner/in, der/die das Projekt begleitet und bei Nachfragen zur Verfügung steht.
- Während der Implementation finden regelmäßige Treffen mit CeDiS statt (ggf. auch per Video-Konferenz), um den Fortlauf der Arbeit zu besprechen.
- Die Arbeiten sollen auf einem GitHub-Repositorium erfolgen, das CeDiS während der Entwicklung einsehen kann.
- Der Auftragnehmer/die Auftragnehmerin fertigt eine technische Dokumentation des Plugins an, die auf dem GitHub-Repositorium abgelegt wird, auf dem die Arbeiten erfolgen.

Technische Rahmenbedingungen

- Das Plugin wird für die Version OJS 3.1 unter PHP 5.6 und höher und MySQL Version 14.14 und höher implementiert.

Bestimmungen über die Ausführungsfrist:

Der Realisierungszeitraum beginnt nach der Auftragsvergabe. Die Realisierung muss spätestens am 15.05.2019 abgeschlossen sein.

Abnahme:

Für die Abnahme der Lieferung/Leistung erfolgt im Rahmen einer Ergebnispräsentation und abschließenden Diskussion. Es gilt folgende besondere Regelung:

- Im Rahmen der Implementierung findet obligatorisch eine Abstimmung zu technischen Aspekten mit CeDiS statt. Diese technisch-funktionelle Abnahme ist entsprechend im Zeitplan zu berücksichtigen.
- Die Gesamt-Abnahme der geleisteten Arbeit wird – nach erfolgter fachlicher Abnahme– explizit vom Auftraggeber innerhalb von vier Kalenderwochen nach Bereitstellung erklärt, sofern keine wesentlichen Mängel vorliegen. Wesentliche Mängel sind solche, welche die zweckmäßige und sinnvolle Nutzung der o.g. Leistungen verhindert.

Fristen:

- Angebotsabgabe bis zum 29.03.2019

Nicht berücksichtigte Angebote:

Das Angebot ist nicht berücksichtigt worden, wenn bis zum Ablauf einer Woche nach Angebotsabgabefrist kein Auftrag erteilt wurde.

Sonstige Informationen:

- Eigenerklärungen nach deutschem Recht sind rechtsverbindlich zu unterzeichnen.
- Im Angebot ist explizit anzugeben, ob der Anbieter/die Anbieterin zur Umsatzsteuer veranlagt wird.
- Im Angebot ist das für den Anbieter/die Anbieterin zuständige Finanzamt sowie die Steuernummer bzw. die Steuer-ID anzugeben.
- Für die TAP sind im Angebot einzelne Beträge/Preise anzugeben.
- Der Auftragnehmer/die Auftragnehmerin muss über eine ausreichende technische Arbeitsumgebung (Hard- und Software) zur Erstellung der digitalen Materialien verfügen.
- Das Angebot sollte einen groben Zeitplan enthalten. In jedem Fall ist aber ein Datum zur Fertigstellung der Leistungen anzugeben.
- Alle im Angebot angegebenen Kosten verstehen sich inkl. evtl. Reisekosten, Spesen und sonstiger Kosten.

- Der Auftragnehmer/die Auftragnehmerin nimmt bei Bedarf und nach inhaltlicher und terminlicher Abstimmung mit dem Auftraggeber an Sachbesprechungen teil. Eine solche Teilnahme kann auch durch entsprechende audiovisuelle Konferenz-Technologien erfolgen.
- Während der Bearbeitung des Auftrags können nach Abstimmung im Detail abweichende Festlegungen getroffen werden.
- Es ist geplant, seitens des Auftraggebers das softwaretechnische Ergebnis des ggf. auf Basis dieser Ausschreibung entstehenden Werks (Software) der Open-Source-Software OJS als frei verfügbare Programmkomponenten hinzuzufügen. Unabhängig davon erhält mit vollständiger Zahlung der vereinbarten Vergütung der Auftraggeber an den vom Auftragnehmer/der Auftragnehmerin erbrachten Arbeitsergebnissen das ausschließliche, zeitlich und räumlich unbegrenzte Recht, die Arbeitsergebnisse für eigene Anwendungen und Zwecke einzusetzen, zu vervielfältigen, zu bearbeiten sowie mit anderen Programmen oder Materialien zu verbinden. Der Auftragnehmer/die Auftragnehmerin überlässt dem Auftraggeber die erstellte Software im Source-Code (siehe oben zu den Technischen Rahmenbedingungen) einschließlich Dokumentation. Nutzungsrechte an Softwareprodukten Dritter, die im Rahmen der Vertragsdurchführung von dem Auftragnehmer/der Auftragnehmerin genutzt und ggf. bearbeitet werden, werden in dem vom Hersteller zugelassenen Umfang übertragen. Der Auftragnehmer/die Auftragnehmerin wird vor der Nutzung nicht eigener Software, welche bei der späteren Verwendung der Ergebnisse des Auftrags durch den Auftraggeber dessen Nutzungs- und Verwertungsrechte einschränken, die Erlaubnis zur Verwendung dieser fremden Software vom Auftraggeber einholen.