

## PEMBANGUNAN APLIKASI NEXTIME KALENDAR PINTAR

<sup>1</sup>Joshua Koh, <sup>1</sup>Zulaiha Ali Othman

Fakulti Teknologi & Sains Maklumat  
43600 Universiti Kebangsaan Malaysia

### Abstrak

Kebanyakan platform kalendar tradisional masih bergantung kepada input manual yang kaku, menjadikan proses penjadualan sesuatu yang membosankan dan tidak efisien. Objektif projek ini adalah untuk menangani masalah input manual yang tidak perlu dengan mencadangkan satu sistem kalendar pintar, iaitu NexTime, yang memanfaatkan kemajuan terkini dalam bidang kecerdasan buatan untuk mentafsir input bahasa semula jadi dan suara bagi tujuan penciptaan acara. Metodologi projek ini menggunakan Model Spiral untuk membolehkan prototaip dibangunkan dengan cepat serta memudahkan perubahan keperluan jika diperlukan. Sistem ini dibina menggunakan teknologi seperti Node.js, dan backend dijalankan di pelayan atas talian. Kebanyakan platform kalendar tradisional masih bergantung kepada input manual yang kaku, menjadikan proses penjadualan sesuatu yang membosankan dan tidak efisien. Objektif projek ini adalah untuk menangani masalah input manual yang tidak perlu dengan mencadangkan satu sistem kalendar pintar, iaitu NexTime, yang memanfaatkan kemajuan terkini dalam bidang kecerdasan buatan untuk mentafsir input bahasa semula jadi dan suara bagi tujuan pembinaan acara. Metodologi projek ini menggunakan Model Spiral untuk membolehkan prototaip dibangunkan dengan cepat serta memudahkan palarasan keperluan jika diperlukan. Sistem ini dibina menggunakan teknologi seperti Node.js, dan backend dijalankan di pelayan atas talian. Kesimpulannya, NexTime mampu mengubah sistem kalendar tradisional kepada satu aplikasi yang lebih pintar dan mesra kepada pengguna, serta membuka laluan kepada penambahbaikan masa hadapan dalam alat produktiviti berasaskan kecerdasan buatan.

Kata Kunci:

**Abstract**

Most traditional calendar platforms rely on rigid manual inputs, making the scheduling process tedious and inefficient. This objective of this project is to addresses the problem of needless manual input of events by proposing a smart calendar system, NexTime, which utilizes the recent advancements of artificial intelligence to interpret natural language and voice input for event creation. The project methodology will be using the Spiral Model for fast prototyping and deployment if there is a need for requirement adjustments. The system will be constructed with tools like Node.js and the backend will be run on an online server. Compared to traditional calendar systems, the suggested system will be able to utilize large language models through APIs that will allow natural language processing to be implemented. The resulting product is a web application using the client-server model for a calendar application that can parse natural language into data that can be used to add events directly to the calendar. This natural language processing was powered by Google's Gemini API. In conclusion, NexTime can transform the traditional calendar application into a smarter, more user-friendly experience and paves the way for future enhancement in AI-driven productivity tools.

Keywords:

**1.0 PENGENALAN**

Pengurusan masa yang berkesan merupakan tunjang utama dalam mencapai produktiviti optimum, baik dalam kerjaya mahupun kehidupan peribadi. Bagi pelajar universiti, cabaran mengimbangi tugas akademik, aktiviti kokurikulum, dan tanggungjawab harian sering mengakibatkan tekanan yang melampau, terutamanya apabila kaedah perancangan tradisional gagal memenuhi keperluan dinamik zaman moden. Menurut kajian awal di Universiti Kebangsaan Malaysia (UKM), lebih 68% pelajar melaporkan bahawa kesukaran mengurus jadual secara manual menyumbang kepada peningkatan tahap stres, sekaligus menjejaskan prestasi akademik dan kualiti hidup (Helmi & Aun, 2023). Situasi ini memperlihatkan keperluan mendesak untuk satu penyelesaian teknologi yang bukan sahaja automatik tetapi juga mampu memahami konteks yang kompleks dalam kehidupan pelajar.

Pada masa ini, kebanyakan alat pengurusan masa seperti planner fizikal atau aplikasi kalendar digital masih bergantung pada input pengguna yang berulang dan struktur jadual yang kaku. Kekurangan fleksibiliti ini menyukarkan pengguna untuk menyesuaikan diri dengan perubahan, seperti tarikh akhir yang dipercepatkan atau pertindihan aktiviti. Akibatnya, proses merancang jadual menjadi suatu beban tambahan yang memerlukan masa dan tenaga yang

signifikan. Di sinilah NexTime, sebuah sistem kalender pintar berasaskan Model Bahasa Besar (LLM) diusulkan sebagai alternatif revolusioner.

NexTime direka untuk memanfaatkan keupayaan semula jadi LLM dalam memahami dan memproses bahasa manusia. Berbeza dengan sistem konvensional yang memerlukan model pembelajaran mesin (ML) tersuai, NexTime menggunakan LLM sedia ada melalui integrasi API. Sebagai contoh, apabila pengguna menulis “Discussion preparation with the group this Friday at 2pm” LLM akan secara automatik mengenal pasti elemen kritikal seperti jenis tugas, tarikh akhir, dan keutamaan. Seterusnya, sistem ini akan menganalisis jadual sedia ada pengguna dan mencadangkan slot masa yang sesuai tanpa pertindihan.

Projek ini dijangka memberikan impak yang signifikan kepada masyarakat dengan mengurangkan beban kognitif berkaitan pengurusan masa, pengguna boleh menumpukan lebih banyak tenaga kepada pembelajaran aktif dan penglibatan sosial. Pada masa yang sama, ciri adaptif NexTime menyokong kesejahteraan mental dengan mengelakkan jadual terlalu padat dan memperuntukkan masa rehat yang mencukupi.

## 2.0 KAJIAN LITERATUR

Satu kajian oleh Michigan State University (2016) mendapati bahawa pelajar sering menghabiskan terlalu banyak masa untuk mencatat tarikh akhir tugas dan tarikh peperiksaan pada kalendar digital, yang boleh menjadi tidak produktif. Walaupun kalendar digital digunakan secara meluas untuk meningkatkan pengurusan masa, kesan sampingan seperti keperluan berterusan untuk menambah dan menyesuaikan tarikh akibat jadual yang kerap berubah boleh secara tidak sengaja mengambil lebih banyak masa daripada yang dijimatkan. Selain itu, penyelidikan oleh Valente, Dominguez-Lara, dan Lourenco (2024) menekankan bahawa kekurangan motivasi dan penangguhan adalah sebab utama mengapa pengguna meninggalkan kalendar digital mereka. Kajian terdahulu mengenai penjadualan digital dan pengurusan masa menunjukkan bahawa walaupun jadual berstruktur memberi manfaat kepada pelajar dan profesional, sistem yang kaku ini sering gagal menampung ketidaktentuan kehidupan sebenar.

Pada masa kini, kalendar digital seperti Google Calendar dan Microsoft Outlook mendominasi pasaran, menawarkan ciri penjadualan asas. Platform lanjutan seperti Reclaim.ai dan Todoist menyediakan fungsi tambahan seperti automasi tugas dan time-blocking, yang

secara automatik memperuntukkan masa untuk tugas sebelum tarikh akhir. Walau bagaimanapun, beberapa ciri seperti penambahan tugas melalui pengecaman suara, boleh menjadi tidak fleksibel dalam keadaan tertentu dan mungkin tidak sesuai untuk semua pengguna. Ciri-ciri lanjutan ini juga mungkin kelihatan tidak mesra kepada pemula yang baru mula mengurus masa dengan lebih berkesan.

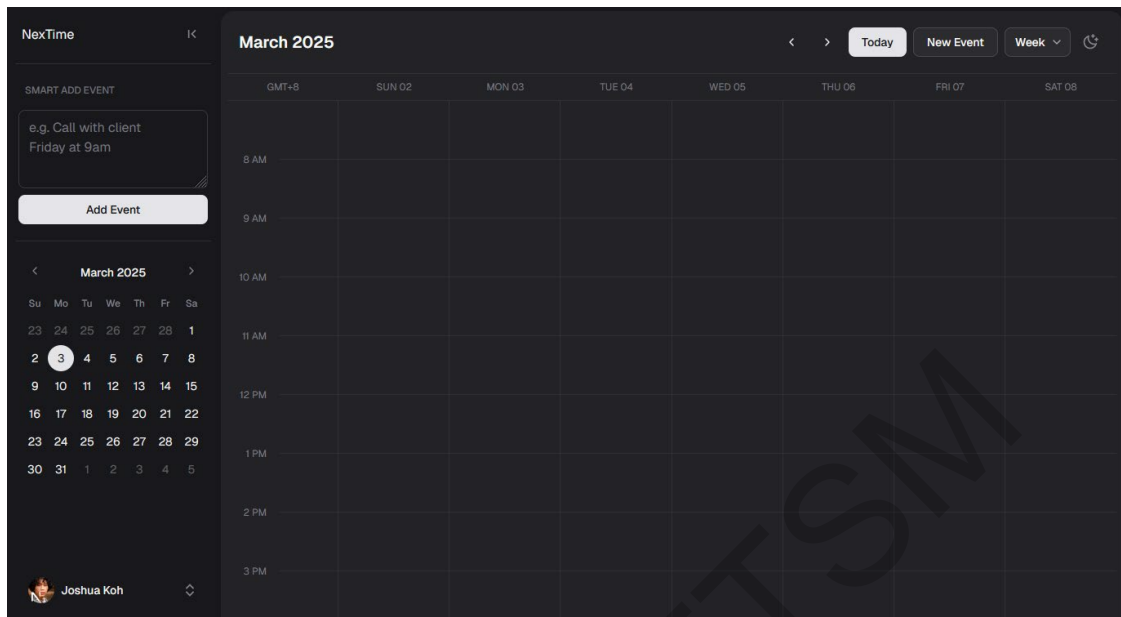
Aplikasi kalendar semasa, seperti Google Calendar, menawarkan kesederhanaan dan kemudahan penggunaan tetapi kekurangan integrasi teknologi canggih untuk menangani konflik jadual dengan berkesan. Sebaliknya, platform maju seperti Reclaim.ai menggabungkan ciri-ciri yang canggih tetapi sering dihalang oleh keluk pembelajaran yang tinggi, yang menghadkan kebolehaksesannya kepada pengguna yang lebih meluas. Kajian terkini telah menekankan potensi algoritma berasaskan AI untuk meningkatkan kecekapan penjadualan, namun penggunaannya masih berada di peringkat awal. Penemuan ini menegaskan keperluan untuk aplikasi kalendar baharu yang menggabungkan kekuatan penyelesaian sedia ada sambil memanfaatkan kemajuan teknologi terkini. Aplikasi sebegini dapat merapatkan jurang yang ada, menawarkan alat pintar dan mesra pengguna untuk memperbaiki pengurusan masa mereka.

### **3.0 METODOLOGI**

Kajian ini merangkumi analisis keperluan, merangka reka bentuk model konseptual, pembangunan aplikasi, pengujian kebolehgunaan dan hasil. Metodologi menerangkan kaedah bagi mengatasi masalah yang dikenal pasti serta menerangkan kajian yang dilakukan.

#### **3.1 Reka Bentuk Antara Muka**

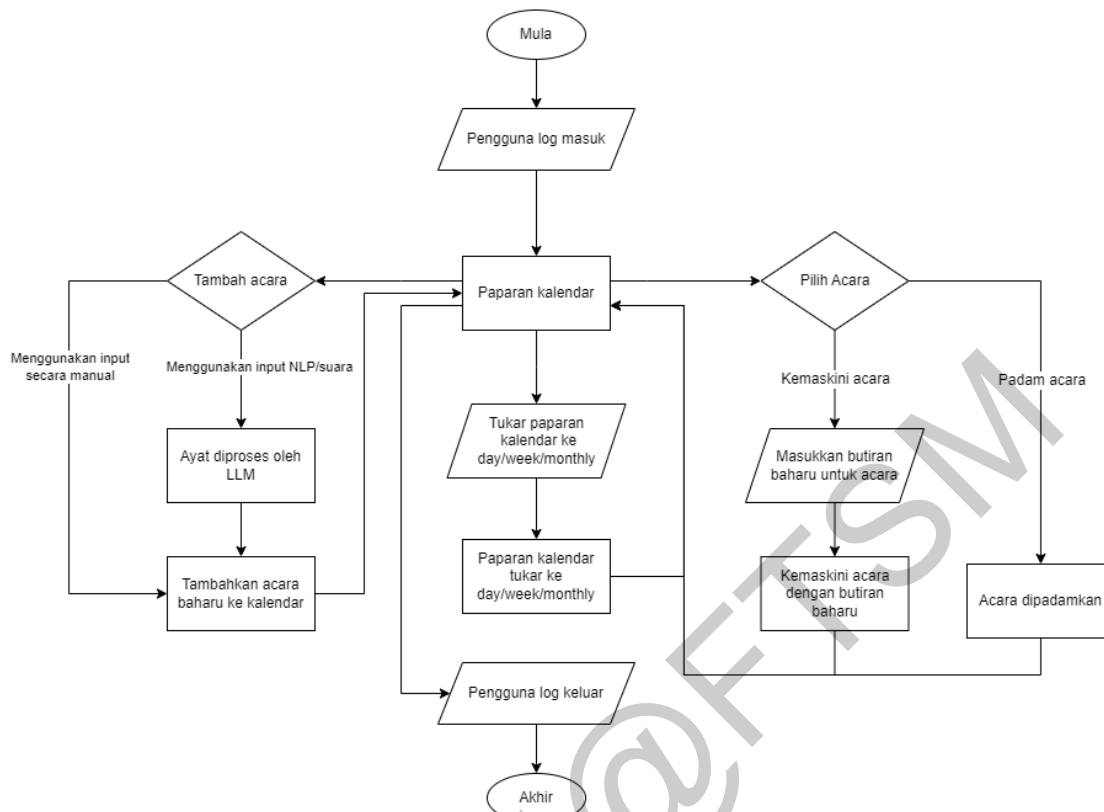
Prototaip dilaksanakan bagi memperoleh keperluan pengguna dengan melakar prototaip ringkas. Teknik ini telah dilakukan untuk memberi gambaran tentang aplikasi yang dibangun. Lakaran antara muka aplikasi dibangun terlebih dahulu menggunakan Figma. Rajah 1 merupakan prototaip ringkas bagi memperoleh keperluan pengguna.



*Rajah 1: Prototaip Antara Muka Kalendar*

### 3.2 Algoritma

Algoritma ini menerangkan urutan logik tindakan yang berlaku apabila pengguna berinteraksi dengan sistem — daripada log masuk sehingga pengurusan acara dalam kalendar. Ia direka bentuk untuk memastikan aliran kerja yang sistematik, responsif dan intuitif, dengan menyepadukan teknologi seperti pemprosesan bahasa semula jadi (NLP) dan input suara yang diproses melalui model bahasa besar (LLM). Dengan algoritma ini, sistem dapat menerima pelbagai bentuk input, mengurus data pengguna dengan cekap, serta membolehkan operasi asas seperti tambah, ubah, dan padam tugas dilakukan dengan mudah. Rajah 2 menunjukkan carta alir algoritma sistem NexTime.



Rajah 2: Carta Alir Algoritma

Proses bermula apabila pengguna log masuk ke dalam sistem. Setelah berjaya log masuk, pengguna akan dibawa ke paparan utama kalendar. Di sini, pengguna mempunyai beberapa pilihan, antaranya ialah menambah acara baharu, mengubah paparan kalendar, atau memilih acara sedia ada untuk dikemaskini atau dipadam.

Untuk menambah acara, pengguna boleh memilih untuk memasukkan butiran secara manual atau menggunakan input bahasa semula jadi atau suara. Jika pengguna memilih input manual, acara akan ditambah ke kalendar secara terus berdasarkan maklumat yang dimasukkan. Sebaliknya, jika input dilakukan melalui bahasa semula jadi atau suara, ayat yang diberikan akan diproses oleh model bahasa besar (LLM) bagi mengenal pasti maklumat penting seperti tarikh, masa dan tajuk acara. Setelah diproses, sistem akan menambah acara baharu tersebut ke dalam kalendar secara automatik.

Selain menambah acara, pengguna juga boleh mengubah paparan kalendar mengikut keperluan mereka, sama ada dalam bentuk harian (day), mingguan (week), atau bulanan (monthly). Fungsi ini membolehkan pengguna melihat tugas mereka dengan lebih fleksibel dan mengikut konteks masa yang sesuai.

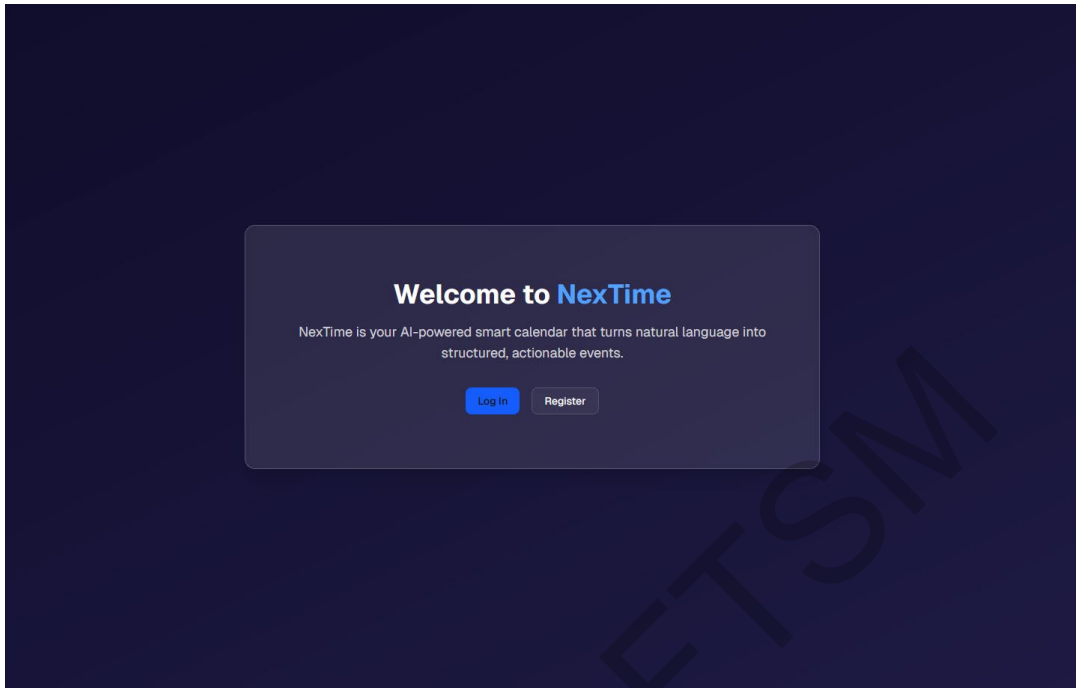
Jika pengguna ingin mengemaskini acara sedia ada, mereka boleh memilih acara tersebut daripada paparan kalendar. Seterusnya, sistem akan meminta pengguna untuk memasukkan butiran baharu yang ingin dikemaskini. Setelah maklumat tersebut dimasukkan, acara akan dikemaskini dengan butiran baharu yang diberikan. Sebagai alternatif, pengguna juga boleh memilih untuk memadam acara, di mana sistem akan terus menghapuskan acara tersebut daripada kalendar.

Akhir sekali, setelah selesai melakukan sebarang operasi, pengguna boleh memilih untuk log keluar daripada sistem. Ini menandakan pengakhiran sesi dan proses operasi dalam carta alir.

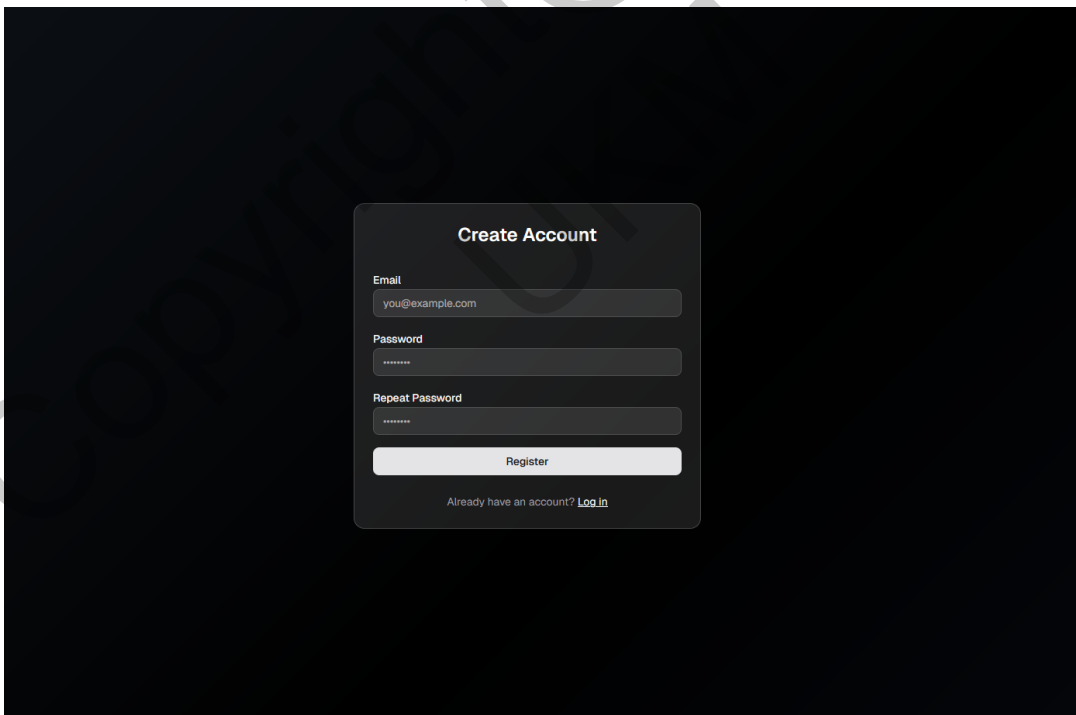
## 4.0 HASIL

### 4.1 Pembangunan Aplikasi

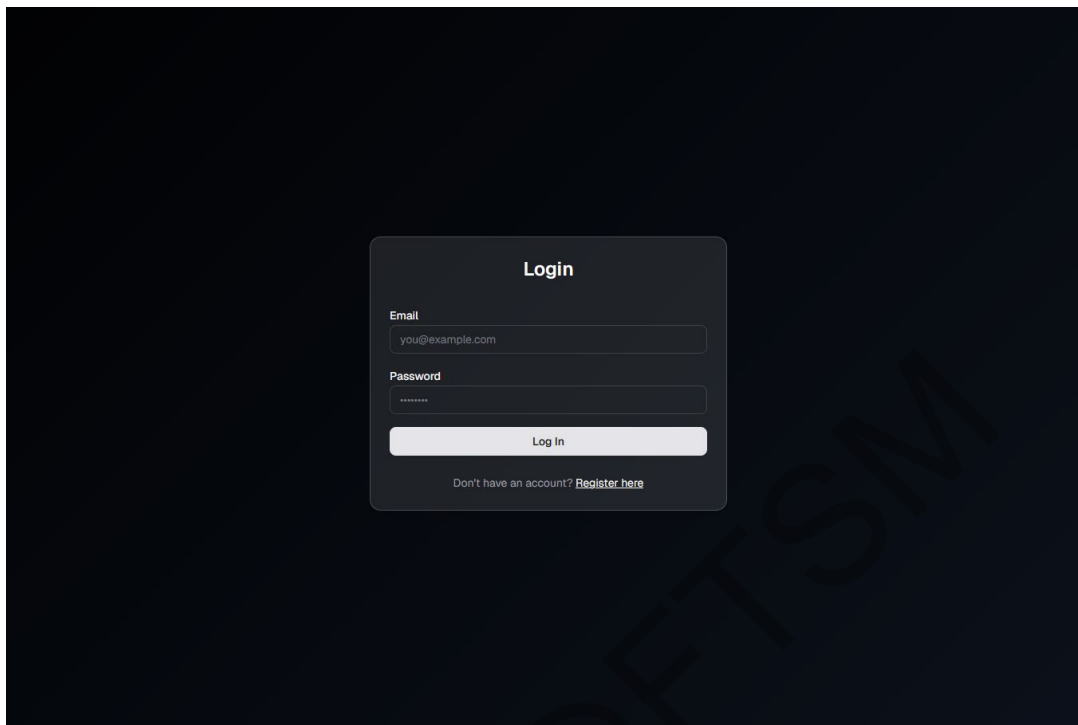
Pada fasa pembangunan, beberapa perisian digunakan bagi menghasilkan aplikasi ini daripada fasa reka bentuk sehingga fasa pembangunan. Keseluruhan aplikasi ini menggunakan bahasa pengaturcaraan *TypeScript* yang menggunakan perisian *NodeJS*. Selain itu, pangkalan data menggunakan *MongoDB* untuk menyimpan maklumat pengguna. Aplikasi ini menggunakan seni bina *Client-Server*. Pangkalan data *MongoDB* digunakan sebagai pangkalan data dalam sistem NexTime kerana sifatnya yang fleksibel, tidak berstruktur (*NoSQL*), dan sesuai untuk aplikasi yang ingin berkembang dengan pantas serta menyimpan pelbagai jenis data seperti teks, tarikh dan masa. Pangkalan data ini menyimpan dokumen dalam bentuk *JSON-like (BSON)* dan membolehkan hubungan antara koleksi dikendalikan secara eksplisit dalam aplikasi. Pustaka *mongoose*” digunakan untuk menghubungkan *MongoDB* dan *NodeJS*. Bahagian *frontend* sistem NexTime dibina menggunakan *NextJS* yang membolehkan pembangunan aplikasi web yang pantas, mesra pengguna, dan responsif. Reka bentuk antara muka disesuaikan untuk memudahkan pengguna menjadualkan acara, mengurus kalendar peribadi dan menerima maklum balas segera daripada sistem. Rajah 3 hingga Rajah 6 merupakan antara muka aplikasi NexTime. Selepas pengguna berjaya log masuk ke dalam aplikasi, aplikasi akan memaparkan antara muka paparan kalendar.



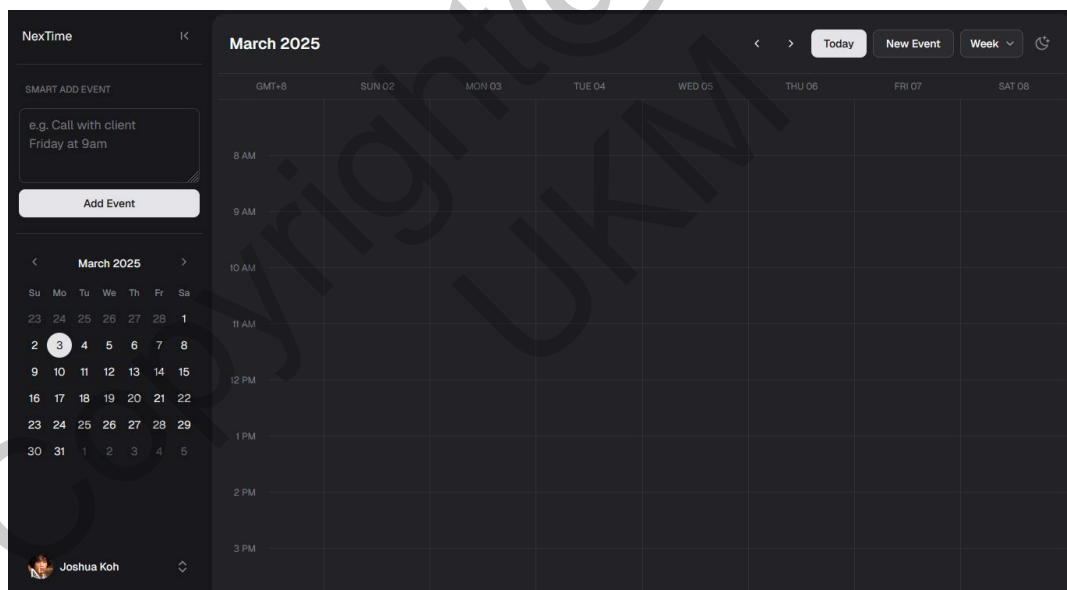
Rajah 3: Antara Muka Halaman Pendaratan



Rajah 4: Antara Muka Daftar Masuk



*Rajah 5: Antara Muka Log Masuk*



*Rajah 6: Antara Muka Paparan Kalendar*

## 5.0 KESIMPULAN

Projek NexTime telah berjaya dilaksanakan sebagai satu aplikasi kalendar pintar yang memanfaatkan teknologi pemprosesan bahasa semula jadi (NLP) untuk memudahkan

pengguna dalam menjadualkan acara harian mereka. Melalui antara muka yang mesra pengguna serta sokongan daripada model bahasa besar (LLM), sistem ini membolehkan pengguna menambah acara dalam kalendar hanya melalui input ayat biasa seperti “Meeting tomorrow at 10 in the morning” atau “Classes this Sunday 9am”. Sepanjang proses pembangunan, sistem ini telah dibina berasaskan seni bina klien-pelayan, menggunakan teknologi moden seperti NextJS untuk bahagian frontend dan Node.js, Express.js serta MongoDB untuk bahagian backend dan penyimpanan data.

Melalui fasa pengujian, didapati bahawa sistem berfungsi dengan baik dari sudut kefungsiian asas seperti pendaftaran dan log masuk pengguna, pengurusan acara, dan integrasi NLP. Ujian penerimaan pengguna (UAT) yang dijalankan menunjukkan majoriti pengguna berpuas hati dengan pengalaman penggunaan sistem, di mana antara muka dianggap mudah difahami, kemas dan sesuai untuk digunakan.

### **5.1 Kekuatan dan Kekangan**

Namun begitu, sistem ini juga mempunyai beberapa kekangan yang wajar diberi perhatian. Antaranya ialah masa respons yang agak perlahan semasa memproses input teks untuk menjana acara menggunakan NLP. Hal ini berpunca daripada keperluan untuk menghantar permintaan ke pelayan LLM luaran, yang memerlukan masa tambahan untuk mendapatkan maklum balas. Di samping itu, fungsi input suara hanya menyokong pelayar berasaskan Chromium seperti Google Chrome dan Microsoft Edge, kerana pustaka yang digunakan tidak disokong dalam Mozilla Firefox, sekali gus menghadkan kebolehecapaian ciri ini kepada sesetengah pengguna. Tambahan pula, penggunaan LLM seperti ChatGPT memerlukan langganan berbayar, menyebabkan projek ini beralih kepada penggunaan model Google Gemini yang lebih sesuai dalam konteks sumber terbuka dan percuma.

### **5.2 Cadangan Penambahbaikan**

Sebagai cadangan untuk penambahbaikan pada masa hadapan, sistem ini boleh diperluaskan dengan beberapa ciri tambahan seperti integrasi kalendar luaran bagi membolehkan penyelarasan acara secara automatik antara pelbagai platform. Di samping itu, sistem input suara boleh ditambahbaik untuk menyokong semua pelayar seperti Mozilla Firefox. Sistem

juga boleh dipertingkat dengan menambah fungsi sistem notifikasi dan pengesahan konflik antara acara secara automatik.

## 6.0 PENGHARGAAN

Penulis kajian ini ingin ucapkan setinggi-tinggi penghargaan dan jutaan terima kasih kepada Prof. Dr. Zulaiha binti Ali Othman, penyelia penulis kajian ini yang telah memberi tunjuk ajar serta bimbingan untuk menyiapkan projek ini dengan jayanya.

Penulis kajian ini juga ingin mengucapkan terima kasih kepada semua rakan sepasukan, keluarga dan rakan-rakan yang memberikan sokongan moral dan motivasi sepanjang menjalankan projek ini. Segala bantuan yang telah dihulurkan amatlah dihargai kerana tanpa bantuan mereka, projek ini tidak dapat dilaksanakan dengan baik.

## 7.0 RUJUKAN

Anand, K., Prasath, R. A., & Bharathvasan, R. K. (2024). College Annual Academic Calendar using Mobile Application.  
<https://doi.org/10.1109/ICCSP60870.2024.10543962>

Commvault. (2025). System Requirements for MongoDB.  
[https://documentation.commvault.com/11.20/system\\_requirements\\_for\\_mongodb.html](https://documentation.commvault.com/11.20/system_requirements_for_mongodb.html)

European Union Digital Library. (2011). A Smart Calendar Application for Mobile Environments. <http://dx.doi.org/10.4108/ICST.MOBIMEDIA2007.1722>

GeeksForGeeks. (2024). Waterfall Model - Software Engineering.  
<https://www.geeksforgeeks.org/waterfall-model/>

GeeksForGeeks. (2024). What is Systems Design – Learn System Design.  
<https://www.geeksforgeeks.org/what-is-system-design-learn-system-design/>

Grzywacz, L. (2024). 2024's Best Calendar Apps for Each Type of Productive People.

<https://lukasz-grzywacz.medium.com/top-calendar-apps-of-2024-find-the-perfect-planner-for-your-needs-ba553ab39b83>

Guinness, H. (2024). The 5 Best Calendar Apps in 2025.

<https://zapier.com/blog/best-calendar-apps/>

Helmi, S. N. B. R., & Aun, N. S. M. (2023). Preliminary Study: Time Management Skills and Stress Levels among Universiti Kebangsaan Malaysia Students. International Journal of Academic Research in Progressive Education and Development.

Inflectra. (2022). What Are System Requirements Specifications? What are System Requirements?

<https://www.inflectra.com/ideas/system-requirements.aspx>

Isa, R., Othman, S., Ali, A. S., Azizan, N., Abdul Aziz, F. F., & Ferguson, J. (2024). FYPMS User Acceptance Testing: Ensuring User Satisfaction and System Success. Journal of Advanced Research in Applied Sciences and Engineering Technology, 59(4), 191-189.

<https://doi.org/10.37934/araset.59.4.181189>

Mozilla. (2024). Firefox System Requirements.

<https://www.mozilla.org/en-US/firefox/121.0/system-requirements/>

Nathan, V. H. (2024). 15 User Acceptance Testing Templates

<https://marker.io/blog/user-acceptance-testing-template>

Norman, D. A. (2013). The Design of Everyday Things. Basic Books.

Nomura, Y., Mihara, S., & Taniguchi, H. (2011). A Practical Calendaring System Conforming with Ambiguous Pattern of Recurring Tasks.

<https://doi.org/10.1109/NBiS.2011.91>

Nomura, Y., Murata, Y., Taniguchi, H., Urata, M., & Muto, S. (2013). Bring Your Own Calendar: A CalDAV-Based Virtual Calendar System.  
<https://doi.org/10.1109/BWCCA.2013.95>

Nordby, K., Løkken, R., & Pfuhl, G. (2019). Playing a video game is more than mere procrastination. *BMC Psychology*.  
<https://link.springer.com/article/10.1186/s40359-019-0309-9>

Oke, A., Aliu, J., Mwanaumo, E., & Abayomi, T. (2023). Leveraging gamification to enhance productivity and employee engagement in the Nigerian construction industry.  
<https://www.emerald.com/insight/content/doi/10.1108/BEPAM-04-2023-0064/full/html>

Onix R. (2024). Release React Native 0.75.  
[https://medium.com/@onix\\_react/release-react-native-0-75-7272b47ceac3](https://medium.com/@onix_react/release-react-native-0-75-7272b47ceac3)

OpenAI. (2025). ChatGPT (Version 4.0) [Large Language Model]. OpenAI.  
<https://chat.openai.com/>

Post, C., Sarala, R., Gatrell, C., & Prescott, J. E. (2020). Advancing theory with review articles. *Journal of Management Studies*, 57(2), 351–376.

Shahrulnizam Bakar. (2013). Kajian Terhadap Corak Pengurusan Masa dan Hubungannya Dengan Prestasi Akademik Para Pelajar Politeknik Melaka.  
[https://www.academia.edu/37141887/Kajian\\_Terhadap\\_Corak\\_Pengurusan\\_Masa\\_dan\\_Hubungannya\\_Dengan\\_Prestasi\\_Akademik\\_Para\\_Pelajar\\_Politeknik\\_Melaka](https://www.academia.edu/37141887/Kajian_Terhadap_Corak_Pengurusan_Masa_dan_Hubungannya_Dengan_Prestasi_Akademik_Para_Pelajar_Politeknik_Melaka)

Srirama, S. N., Flores, H., & Paniagua, C. (2011). Zompopo: Mobile Calendar Prediction Based on Human Activities Recognition Using the Accelerometer and Cloud Services.  
<https://doi.org/10.1109/NGMAST.2011.21>

Stefanidi, E. (2023). Literature Reviews in HCI: A Review of Reviews.

[https://www.researchgate.net/publication/368881869\\_Literature\\_Reviews\\_in\\_HCI\\_A\\_Review\\_of\\_Reviews](https://www.researchgate.net/publication/368881869_Literature_Reviews_in_HCI_A_Review_of_Reviews)

*Joshua Koh (A194821)*

*Prof. Dr. Zulaiha Binti Ali Othman*

Fakulti Teknologi & Sains Maklumat

Universiti Kebangsaan Malaysia